Underwater Acoustic OFDM: Algorithm Design, DSP Implementation, and Field Performance

Lei Wan
pires810@gmail.com

Follow this and additional works at: https://opencommons.uconn.edu/dissertations

Recommended Citation
https://opencommons.uconn.edu/dissertations/378
Multicarrier modulation in the form of orthogonal frequency division multiplexing (OFDM) has now been recognized as an appealing solution for high data rate communications over underwater acoustic channels with large delay spread. This dissertation covers three research topics: (i) transceiver algorithm design, (ii) real-time transceiver implementation and optimization on DSP platforms, and (iii) analysis of environmental impact on communication performance, unified under a common objective of bringing the underwater acoustic OFDM technology into practical systems.

On algorithm design for underwater acoustic OFDM, this thesis first investigates a key receiver module of Doppler scale estimation. Underwater acoustic channels are fast-varying due to the platform motions and the medium instability, so that the received waveforms are often stretched or compressed, rendering large Doppler shifts on OFDM subcarriers. Accurate Doppler scale estimation, and the subsequent Doppler compensation, are of vital importance for robust performance in practical channels. In this thesis, we compare various Doppler scale estimators exploiting different OFDM signal structures, in both single-input single-output
(SISO) and distributed multiple-input multiple-output (MIMO) systems. Second, this thesis investigates adaptive modulation and coding (AMC) for underwater acoustic OFDM, where the transmitter parameters adapt to time-varying channel conditions. We develop an AMC-OFDM system based on a finite number of transmission modes, either to improve the system throughput under a fixed transmission power or to reduce the energy consumption for a finite-length data packet. The proposed AMC-OFDM operation is verified in sea experiments.

For DSP-based implementation, we first optimize the receiver algorithms to achieve real-time receiver processing. In this thesis, we consider two OFDM modem prototypes, one with a single transmitter and a single receiver, and the other with two transmitters and two receivers, and pursue both floating- and fixed-point implementations. Second, we implement an OFDM-modulated dynamic coded cooperation (DCC) in a three-node network with a source, a destination, and a relay. The relay can superimpose its transmission to the ongoing transmission from the source to the destination, after it decodes the message from the source correctly.

Finally, this thesis analyzes the performance of underwater OFDM modems in a recent two-month deployment in the Chesapeake Bay. We correlate the receiver performance with environmental parameters, e.g., wind speed and wave height, and also explore advanced offline receiver algorithms to process data sets
that failed decoding during online operations. This study enables a good understanding of the environmental impact on the underwater OFDM performance, and provides guidelines on the selection of modem parameters.
Underwater Acoustic OFDM: Algorithm Design, DSP

Implementation, and Field Performance

Lei Wan

Department of Electrical & Computer Engineering

University of Connecticut

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of Doctor of Philosophy

at the

University of Connecticut

2014
Copyright by

Lei Wan

2014
APPROVAL PAGE

Doctor of Philosophy Dissertation

Underwater Acoustic OFDM: Algorithm Design, DSP
Implementation, and Field Performance

Presented by

Lei Wan, B.S., M.S.

Major Advisor

Shengli Zhou

Co-major Advisor

Zhijie Shi

Associate Advisor

Peter Willett

University of Connecticut

2014
To my family
ACKNOWLEDGEMENTS

First, I want to express my sincere gratitude to my advisor, Professor Shengli Zhou for his supervision and support in both academia and daily life. Without his constant help during the past 5 years, none of the progresses I made is possible. His broad knowledge and sharp insight always guide me in the right direction.

I also would like to thank Professor Zhijie Shi, Professor Peter Willett, Professor Yaakov Bar-Shalom and Professor Krishna Pattipati. The abundant knowledge I have learned from their classes have greatly helped in my past research and will continue to benefit my future career.

I want to thank my colleagues Jianzhong Huang, Xiufeng Song, Shuo Zhang, Hao Zhou, Xiaoka Xu, Yi Huang, Zhaohui Wang, Haining Mo, and Yibo Zhu. They bring knowledge as well as joy to my life in UCONN.

Finally I must thank my parents for their understanding and constant support.
## TABLE OF CONTENTS

### Chapter 1: Introduction

1.1 Current State-of-the-Art of Underwater Acoustic Communications 5

1.2 Motivation of Doppler Scale Estimation 8

1.3 Motivation of Adaptive Modulation and Coding 9

1.4 Motivation of SISO and MIMO OFDM DSP Optimization 9

1.5 Motivation of Dynamic Coded Cooperation in a Relay Network 10

1.6 Motivation of Performance Analysis for Long Term Deployment 11

1.7 Outline of the Dissertation 11

1.8 Publications 12

### Chapter 2: Basic Transmitter and Receiver Structure

2.1 Signal Structure 15

2.1.1 Transmitted Data Packet 15

2.1.2 Subcarrier Allocation 16

2.2 Channel Model 17

2.3 Receiver Process 18

2.3.1 Time Domain Received Signal 18

2.3.2 Frequency Domain Signal 19

2.3.3 Maximum-Ratio Combining (MRC) 20

2.3.4 Log Likelihood Ratio (LLR) Computation 21

2.4 Connection between This Chapter and Following Chapters 21
Chapter 3: Algorithm Design for Underwater Acoustic OFDM

3.1 Comparison of Doppler Scale Estimation Methods for Underwater Acoustic OFDM

3.1.1 Introduction to Doppler Scale Estimation

3.1.2 Doppler Scale Estimation with A CP-OFDM Preamble

3.1.2.1 Self-Correlation

3.1.2.2 Cross-Correlation

3.1.3 Doppler Scale Estimation with Each ZP-OFDM Block

3.1.3.1 Null-Subcarrier Based Blind Estimation

3.1.3.2 Pilot-Aided Estimation

3.1.3.3 Decision-Aided Estimation

3.1.4 Simulation Results

3.1.4.1 RMSE Performance with CP-OFDM

3.1.4.2 RMSE Performance with ZP-OFDM

3.1.4.3 Comparison of Blind Methods of CP- and ZP-OFDM

3.1.4.4 BLER Performance with ZP-OFDM

3.1.5 Experimental Results

3.1.5.1 Test Case 1

3.1.5.2 Test Case 2

3.1.6 Extension to Distributed MIMO-OFDM

3.1.6.1 Pilot- and Decision-aided Estimation

3.1.6.2 Null-Subcarrier Based Blind Estimation
3.1.7 Summary ..................................................... 44

3.2 Adaptive Modulation and Coding (AMC) for Underwater Acoustic OFDM ................................. 45

3.2.1 Introduction to AMC for Underwater Acoustic OFDM .................................................. 45

3.2.2 Construction of Transmission Modes ................................................................. 47

3.2.2.1 Modulation and Coding Pairs ................................................................. 47

3.2.2.2 Transmission Modes on an OFDM Modem Platform ......................................... 49

3.2.2.3 AMC Procedure ................................................................. 51

3.2.3 Performance Metrics ................................................................. 52

3.2.3.1 Existing Performance Metrics ................................................................. 52

3.2.3.2 Proposed Performance Metric ................................................................. 53

3.2.3.3 Simulation Results ................................................................. 54

3.2.3.4 Performance Results with Recorded Data .................................................. 55

3.2.4 Real Time AMC Tests ................................................................. 58

3.2.4.1 Test 1: AMC Between Nodes 9 and 5 .................................................. 61

3.2.4.2 Test 2: AMC Between Nodes 7 and 5 .................................................. 62

3.2.5 Energy Consumption Optimized AMC ................................................................. 64

3.2.5.1 Problem Formulation ................................................................. 65

3.2.5.2 Numerical Examples ................................................................. 67

3.2.6 Summary ..................................................... 72
Chapter 4: DSP Implementation for Underwater Acoustic OFDM

4.1 SISO and MIMO OFDM DSP Optimization for Underwater Acoustic Modems .............................................. 73

4.1.1 Introduction to SISO and MIMO OFDM DSP Implementation for Underwater Acoustic Modems ............. 73

4.1.2 OFDM Transceiver Design ........................................ 75

4.1.2.1 Transmitter Design ............................................. 75

4.1.2.2 Receiver Overview .............................................. 77

4.1.2.3 Synchronization ................................................. 78

4.1.2.4 Payload Block-by-block Processing ......................... 79

4.1.3 Floating-Point Implementation .................................. 80

4.1.3.1 Hardware Platform ............................................. 80

4.1.3.2 Implementation .................................................. 81

4.1.3.3 Processing Time ................................................ 83

4.1.3.4 BLER Performance ............................................. 85

4.1.4 Fixed-Point Implementation .................................... 87

4.1.4.1 Implementation .................................................. 87

4.1.4.2 Processing Time ................................................ 88

4.1.4.3 BLER Performance ............................................. 89

4.1.5 Performance Results of Replaying Data from a Sea Test .. 91

4.1.5.1 Received SNR Fitting .......................................... 92
4.1.5.2 Channel Impulse Response and Synchronization . . . . 93
4.1.5.3 Decoding Results ........................................ 94
4.1.5.4 Discussions ................................................ 95
4.1.6 Summary ...................................................... 96

4.2 System Implementation of OFDM-DCC in Underwater Acoustic Channels .............................................. 96

4.2.1 Introduction to Dynamic Coded Cooperation in Underwater Acoustic Communications .......................... 96
4.2.2 OFDM Modulated Dynamic Coded Cooperation ....... 98
4.2.2.1 Layered Erasure- and Error-Correction Coding ..... 99
4.2.2.2 Relay Operations ......................................... 101
4.2.2.3 Receiver Processing at the Destination ................. 103
4.2.2.4 Discussions ............................................... 104
4.2.3 Implementation of One OFDM-DCC System ............ 105
4.2.4 Experiment Results of the Implemented System ...... 106
4.2.4.1 Swimming Pool Test .................................... 107
4.2.4.2 Sea Test .................................................... 110
4.2.5 Summary ...................................................... 114

Chapter 5: Field Performance of Underwater Acoustic OFDM 115

5.1 Introduction to the OFDM Modem Deployment in Chesapeake Bay 115

5.2 Online Receiver Performance ................................... 119
## LIST OF TABLES

1. OFDM Parameters in Simulations .......................... 31
2. Modulation and coding pairs for AMC OFDM .............. 48
3. Payload of the 5 modes with $T_g = 50$ ms (Note that 4 bytes reserved by the modem physical layer are excluded in the computation) 50
4. AMC test result (between nodes 9 and 5) .................. 60
5. AMC test result (between nodes 7 and 5) .................. 62
6. Energy consumption related parameters .................... 68
7. ZP-OFDM Parameters ....................................... 76
8. Processing time of one SISO OFDM block of 210 ms ........ 83
9. Processing time of one MIMO OFDM block of 210 ms .... 84
10. Processing Time of Floating- and Fixed-point implementation ... 89
11. Experiment related parameters ............................ 105
12. Data provided by CBIBS .................................. 116
# LIST OF FIGURES

1. The data burst structure considered in this thesis, which consists of a special CP-OFDM preamble and multiple ZP-OFDM blocks. 16

2. Top: Doppler scale estimation with Doppler-insensitive waveforms. Bottom: Doppler scale estimation with Doppler-sensitive waveforms. 25

3. Performance of different estimators for the CP-OFDM preamble in single-path and multipath channels (channel settings 1 and 2). 33

4. Performance of different estimators for ZP-OFDM in single-path channels (channel setting 1). The CRLB with all data known is included as a benchmark. 35

5. Performance of different estimators for ZP-OFDM in multipath channels with a common Doppler scale (channel setting 2). 36

6. Null subcarrier based method in ZP-OFDM and CP-OFDM. 37

7. The BLER performance in multipath multi-Doppler channels (channel setting 3). 38

8. MACE10: Estimated Doppler speeds for 30 data bursts in MACE10, where each data burst has 20 OFDM blocks. The time interval between two consecutive data bursts is around 4 mins. 39

9. Estimated channel impulse responses for two different blocks at different bursts. 39
10 MACE10: Performance comparison of Doppler estimation approaches, file ID: 1750155F1954_0C0_S5. .................................................. 41

11 MACE10: BLER Performance using different Doppler estimation methods by adding artificial noise to the received signal, $\hat{\sigma}^2$ denoting the estimated ambient noise power. .................................................. 41

12 Pilot- and decision-aided Doppler scale estimation in a distributed two-user ZP-OFDM system. .................................................. 43

13 Illustration of the objective functions of the null-subcarrier based method in a two-user system. .................................................. 45

14 BLER performance of the five modulation and coding pairs. .......... 49

15 The data transmission format. .................................................. 50

16 AMC procedure. .................................................. 51

17 BLER for five transmission modes. .................................................. 55

18 The estimated channel scattering function in three different locations. .................................................. 57

19 Performance results using experimental data sets. ...................... 58

20 Locations of the nodes used in the AMC tests. ...................... 59

21 Estimated channel scattering functions in the AMC tests. .......... 59

22 Waveform affected by impulsive noise. .................................................. 60

23 SNR performance for AMC test 1. .................................................. 61

24 SNR performance for AMC test 2. .................................................. 63
The measured instantaneous power consumption at different transmit power levels. 64

Measured data block transmit power. 69

Example 1: the energy consumption optimized AMC. 70

Example 2: the energy consumption optimized AMC. 71

The SISO, SIMO (1x2), and MIMO (2x2) system setups. 75

The procedure of the OFDM receiver. 77

The experimental setup in a water tank. 84

One recorded data burst of one preamble followed by 10 data blocks. 85

Channel impulse responses measured on one receiver. 86

BLER as a function of pilot SNR, SISO-OFDM. 86

BLER as a function of pilot SNR, MIMO-OFDM. 86

The comparison of fixed- and floating-point implementation of SISO-OFDM. 89

The comparison of fixed- and floating-point implementation of MIMO-OFDM, 2 transmitters and 2 receivers. 90

Input SNR fitting. 92

Estimated channel impulse responses on two phones. 93

The correlation plateau for synchronization. 93

Decoding performance of phone 3 and 4. 94

Surface relay system. 98
43 Relay introduces a waiting time prior to the cooperative transmission to achieve the block-level synchronization.

44 Experimental setup in the swimming pool.

45 There is only one relay used between the source and the destination. The relay can be placed at different locations, as marked.

46 Samples of the estimated channels without the relay cooperation and with the relay cooperation.

47 The packet error rate is obtained by adding noise to the recorded data at the destination. Note that the relay operation is done online in real time.

48 The locations of the source (node 4), the relay (node 5) and the destination (node 9).

49 (a) The OFDM modem before entering the water during deployment; (b) The buoy after being deployed.

50 One received waveform after bandpass filtering; there are some impulsive noises.

51 OFDM-DCC sea test results.

52 The estimated channels before and after the relay cooperation.

53 Location of the OFDM modems deployed at the Chesapeake Bay.

54 The transmission protocol between the two modems.

55 Online modem performance; plotted based on the log files.

56 Factors related with PSNR.
# Acronyms and Notations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACK</td>
<td>acknowledgement</td>
</tr>
<tr>
<td>AGC</td>
<td>automatic gain control</td>
</tr>
<tr>
<td>AMC</td>
<td>adaptive modulation and coding</td>
</tr>
<tr>
<td>BER</td>
<td>bit error rate</td>
</tr>
<tr>
<td>BLER</td>
<td>block error rate</td>
</tr>
<tr>
<td>BPSK</td>
<td>binary phase-shift keying</td>
</tr>
<tr>
<td>CFO</td>
<td>carrier frequency offset</td>
</tr>
<tr>
<td>CP</td>
<td>cyclic prefix</td>
</tr>
<tr>
<td>CRLB</td>
<td>Cramer-Rao lower bound</td>
</tr>
<tr>
<td>CTS</td>
<td>clear to send</td>
</tr>
<tr>
<td>DCC</td>
<td>dynamic coded cooperation</td>
</tr>
<tr>
<td>DSP</td>
<td>digital signal processor</td>
</tr>
<tr>
<td>DSSS</td>
<td>direct-sequence spread spectrum</td>
</tr>
<tr>
<td>ESNR</td>
<td>effective signal-to-noise ratio</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>FFT</td>
<td>fast Fourier transform</td>
</tr>
<tr>
<td>FSK</td>
<td>frequency-shift keying</td>
</tr>
<tr>
<td>GF</td>
<td>Galois field</td>
</tr>
<tr>
<td>GPS</td>
<td>global positioning system</td>
</tr>
<tr>
<td>HFM</td>
<td>hyperbolic-frequency-modulated</td>
</tr>
<tr>
<td>IBI</td>
<td>inter-block interference</td>
</tr>
<tr>
<td>ICI</td>
<td>inter-carrier interference</td>
</tr>
<tr>
<td>i.i.d.</td>
<td>identically and independently distributed</td>
</tr>
<tr>
<td>IFFT</td>
<td>inverse fast Fourier transform</td>
</tr>
<tr>
<td>ISNR</td>
<td>input signal-to-noise ratio</td>
</tr>
<tr>
<td>LDPC</td>
<td>low-density parity-check</td>
</tr>
<tr>
<td>LFM</td>
<td>linear-frequency-modulated</td>
</tr>
<tr>
<td>LLR</td>
<td>log likelihood ratio</td>
</tr>
<tr>
<td>LPF</td>
<td>low pass filter</td>
</tr>
<tr>
<td>LS</td>
<td>least squares</td>
</tr>
<tr>
<td>MIMO</td>
<td>multiple-input multiple-output</td>
</tr>
<tr>
<td>MMSE</td>
<td>minimum mean square error</td>
</tr>
<tr>
<td>MRC</td>
<td>maximum-ratio combining</td>
</tr>
<tr>
<td>NOAA</td>
<td>National Oceanic and Atmospheric Administration</td>
</tr>
<tr>
<td>OFDM</td>
<td>orthogonal frequency division multiplexing</td>
</tr>
<tr>
<td>PAPR</td>
<td>peak-to-average power ratio</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Definition</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
</tr>
<tr>
<td>PER</td>
<td>packet error rate</td>
</tr>
<tr>
<td>PSK</td>
<td>phase-shift keying</td>
</tr>
<tr>
<td>PSNR</td>
<td>pilot signal-to-noise ratio</td>
</tr>
<tr>
<td>QAM</td>
<td>quadrature amplitude modulation</td>
</tr>
<tr>
<td>QPSK</td>
<td>quadrature phase-shift keying</td>
</tr>
<tr>
<td>RF</td>
<td>radio frequency</td>
</tr>
<tr>
<td>RMSE</td>
<td>root-mean-square error</td>
</tr>
<tr>
<td>RTS</td>
<td>request to send</td>
</tr>
<tr>
<td>SISO</td>
<td>single-input single-output</td>
</tr>
<tr>
<td>SIMO</td>
<td>single-input multiple-output</td>
</tr>
<tr>
<td>SNR</td>
<td>signal-to-noise ratio</td>
</tr>
<tr>
<td>TI</td>
<td>Texas Instruments</td>
</tr>
<tr>
<td>UWSN</td>
<td>underwater wireless sensor networks</td>
</tr>
<tr>
<td>ZF</td>
<td>zero forcing</td>
</tr>
<tr>
<td>ZP</td>
<td>zero padding</td>
</tr>
</tbody>
</table>
Notation

\( x \) scalar

\( \mathbf{x} \) vector

\( \mathbf{X} \) matrix

\( (\cdot)^* \) conjugate

\( |\cdot| \) absolute

\( \|\cdot\|_n \) \( n \)-th norm

\( * \) convolution
1.1 Current State-of-the-Art of Underwater Acoustic Communications

In underwater environments, acoustic wave is preferred over radio and optical waves as information carrier, because the latter two suffer from high attenuation and severe scattering correspondingly in the medium of water. According to the nature of acoustic wave propagation, in shallow water environment, there are sound reflection at the surface, bottom and any objects; in deep water environment, there are sound refraction caused by speed variation with depth [10, 81]. Both mechanisms will lead to multipath effect in underwater acoustic communications. On top of this multipath effect, the relative low propagation speed of acoustic wave in water ($c = 1500 \text{ m/s}$) often leads to relatively large delay spread. Hence, underwater acoustic communication systems often suffer from frequency selective fading.
On the other hand, in underwater environments, instruments are subject to drifting with waves, currents and tides [81]. So underwater acoustic communication channel is indeed fast varying, and hence often fast fading. When the speeds of the drifting movements are comparable to sound speed $c$, there is significant Doppler effect. The distortion introduced by Doppler effect is often modeled as frequency shifting in narrowband systems; and as waveform dilation or compression in wideband systems.

In order to overcome the challenging underwater acoustic communication channel, there have been significant amount of efforts dedicated on this area since late 1980s. Before orthogonal frequency division multiplexing (OFDM) prevails in recent years, efforts were focused on single carrier communication systems, and remarkable progresses have been made [18, 84, 82, 37, 59, 75, 85, 87, 95, 108, 118]. In these works, different modulation schemes like noncoherent frequency-shift keying (FSK), coherent phase-shift keying (PSK), direct-sequence spread spectrum (DSSS), etc., have been adopted.

In recent years, multi-carrier transmission in the form of OFDM has been very actively pursued for underwater acoustic communications [7, 15, 45, 48, 49, 54, 56, 55, 61, 71, 72, 86, 92, 97, 102, 107, 117, 110, 109, 20, 96, 19, 99, 100]. Compared with single carrier system, OFDM has advantages of high bandwidth efficiency and low receiver complexity. The drawbacks of an OFDM system are its high peak-to-average power ratio (PAPR) value and vulnerability to channel time variations.
In contrast with extensive research work, there are fewer work on underwater acoustic modem development, which include commercial products such as [6, 60, 28], a model widely used in the research community [32], and experimental designs such as [78, 35, 104, 5]. All of these modems are based on single carrier systems. Regarding to underwater acoustic OFDM modem development, there are even fewer work. [114, 91, 112, 111] are among the limited examples in this area.

With all the developed theories and practical modems, people are interested in their performance in real sea environments. Finding out which are the relevant environmental factors and what are their impacts on underwater water acoustic communication systems is critical for future system improvement. A few works have been dedicated on this purpose. [76] discusses the performance of a single carrier binary phase-shift keying (BPSK) decision feedback equalizer (DFE) system during an experiment. In [69], the performance of a single carrier PSK system in a 3-month experiment is presented. In [33], the performance of WHOI Micro-Modem PSK signal system in the RACE and SPACE experiments are presented. In [98], system performance of underwater acoustic OFDM during a two-month deployment is analyzed and correlated with environmental conditions.

This thesis focuses on the algorithm design, digital signal processor (DSP) development and field performance for underwater acoustic OFDM. Specifically, it investigates the following five research problems.

- [3.1] Doppler scale estimation for underwater acoustic OFDM;
• [3.2] Adaptive modulation and coding for underwater acoustic OFDM;

• [4.1] SISO and MIMO OFDM DSP optimization;

• [4.2] Dynamic coded cooperation in a relay network;

• [5] Performance analysis for a two-month deployment in the Chesapeake Bay.

1.2 Motivation of Doppler Scale Estimation

As mentioned previously, an OFDM system enjoys the advantages of high bandwidth efficiency and low receiver complexity. The high bandwidth efficiency property of OFDM is facilitated by the orthogonality among different subcarriers. In case there is frequency offset, this orthogonality will be destroyed and system performance will degrade.

Unfortunately, in underwater acoustic communications, due to the constant movement of medium, and the relatively slow propagation speed of sound wave, there are distinct Doppler frequency offsets in most cases. Hence, Doppler scale estimation is one key receiver module for underwater acoustic communication systems.

In this thesis, we’ll focus on the OFDM signal waveform that has been implemented in [111], which includes both cyclic prefix based OFDM (CP-OFDM) preamble and zero padding based OFDM (ZP-OFDM) data symbols. The CP-OFDM and ZP-OFDM structures provide us plenty of choices for Doppler scale
estimation. We will carry out extensive comparisons among those choices. This problem will be studied in Chapter 3, Section 3.1.

1.3 Motivation of Adaptive Modulation and Coding

As pointed out in [4, 76, 83], underwater acoustic channels are fast varying spatially and temporally. Adaptive modulation and coding (AMC) technique [36, Chapter 8] is appealing for underwater acoustic communications to improve the system efficiency by varying transmission parameters according to channel conditions.

Limited attention has been paid to AMC for underwater acoustic communications. Among these works, most of them do not provide an AMC system that can operate on real time. The work in this thesis will utilize a commercial modem and build several working modes on it. Real time AMC-OFDM sea trials will be carried out to test the system. More details about this problem will be illustrated in Chapter 3, Section 3.2.

1.4 Motivation of SISO and MIMO OFDM DSP Optimization

In recent years, there have been growing interests in building underwater wireless sensor networks (UWSN) [1, 25, 41], in which underwater acoustic communication modems which provide high performance and reliable communication are fundamental elements.
There have been limited amount of work on building underwater acoustic communication modems, most of which focus on single carrier and single-input single-output (SISO) systems. In this thesis, we will build OFDM underwater acoustic communication modem prototypes based on (Texas Instruments) TI DSP platforms, for both SISO and 2×2 multiple-input multiple-output (MIMO) cases. Details about this work will be provided in Chapter 4, Section 4.1.

1.5 Motivation of Dynamic Coded Cooperation in a Relay Network

In challenging underwater acoustic communication channels, relay is an effective way to improve system throughput. Recently, dynamic coded cooperation (DCC) scheme has been proposed and it’s particularly interesting. In the DCC scheme, the half-duplex relay listens until it can decode the message correctly and then switches to the transmission mode. The signal from the relay superimposes on the ongoing transmission from the source to the destination. No extra time is needed for relay transmission, and the source can be even unaware of the existence of the relay.

Existing works on the DCC scheme assume frequency-flat channels, and assume symbol level (or sample level) synchronization of the transmissions from the source and the relay, neither of which is realistic in underwater acoustic communication environments. In this thesis, we’ll implement the DCC scheme on top of an OFDM modulation system, where the repetition redundancy (RR) strategy is applied in the relay node to build an OFDM-DCC system that works in
underwater acoustic environments. This problem will be studied in Chapter 4, Section 4.2.

1.6 Motivation of Performance Analysis for Long Term Deployment

In order to better utilize oceans, there have been growing interests in building up underwater acoustic communication systems and networks. While some applications require short time responses, many applications of underwater acoustic sensor networks require long term deployment and monitoring.

Since March 2012, National Oceanic and Atmospheric Administration’s (NOAA’s) Chesapeake Bay Interpretive Buoy System (CBIBS) has chosen the OFDM modem as the data communication tool, which is the first practical application of the AquaSeNT OFDM modem. In this thesis, we’ll analyze the decoding performance of data sets over a two-month period during this deployment, from April 8, 2013 to June 5, 2013, and then correlate it with environmental conditions to study the impact of environmental conditions on underwater acoustic communications. Based on this study, suggestions on possible performance enhancement for long term deployments in shallow water environments will be provided. This part of work is presented in details in Chapter 5.

1.7 Outline of the Dissertation

Chapter 2 gives a brief overview on the receiver processing for ZP-OFDM with multiple receive elements.
Chapter 3 covers algorithm design for underwater acoustic OFDM, including Doppler scale estimation and adaptive modulation and coding.

Chapter 4 introduces the DSP implementation of underwater acoustic OFDM modem prototypes, and the implementation of a three-node DCC relay system.

Chapter 5 investigates field performance analysis for underwater acoustic OFDM.

Chapter 6 concludes the dissertation.

1.8 Publications

The results from the following publications have been included in this thesis.

Journal papers:


*Conference papers:*


The results from the following papers have not been included in this thesis.

*Journal papers:*


*Conference papers:*


Chapter 2

Basic Transmitter and Receiver Structure

In this chapter, a brief overview on system signal structure, channel model and receiver process will be provided.

2.1 Signal Structure

2.1.1 Transmitted Data Packet

For most work in this thesis, transmitted data packet consists of a CP-OFDM preamble and followed by several ZP-OFDM data blocks. The function of CP-OFDM preamble includes detection, synchronization and Doppler scale estimation [65, 97]. This transmission format, as shown in Fig. 1, has been implemented on DSP-based OFDM modem prototypes [111]. Fig. 1 shows a two transmitter example, which includes the single transmitter system as a special case by turning the second antenna off.
Figure 1: The data burst structure considered in this thesis, which consists of a special CP-OFDM preamble and multiple ZP-OFDM blocks.

In Fig. 1, $T_{\text{pse}}$ is the guard time between CP-OFDM preamble and ZP-OFDM data blocks, $T$ is the ZP-OFDM symbol duration, and $T_{g}$ is the guard time between ZP-OFDM data blocks.

### 2.1.2 Subcarrier Allocation

According to Fig. 1, for CP-OFDM preamble, we can write its baseband symbol as:

$$x_{cp}(t) = \sum_{k=-K_0/2}^{K_0/2-1} s[k] e^{j2\pi \frac{k}{T_0} t} q(t), \quad t \in [-T_{cp}, 2T_0]$$  

where $T_0$ is the symbol duration, $K_0$ is number of subcarriers in one CP-OFDM symbol, $s[k]$ is the transmitted symbol on the $k$-th subcarrier, $T_{cp}$ is the duration of cyclic prefix, and $q(t)$ is the rectangular pulse shaping window.

$$q(t) = \begin{cases} 
1, & t \in [-T_{cp}, 2T_0]. \\
0, & \text{elsewhere}. 
\end{cases}$$  

For ZP-OFDM, its baseband symbols can be written as:

$$x_{zp}(t) = \sum_{k=-K/2}^{K/2-1} s[k] e^{j2\pi \frac{k}{T} t} g(t), \quad t \in [0, T_{bl}]$$
where $K$ is the number of subcarriers, $s[k]$ is the transmitted symbol on the $k$-th subcarrier, $T_{bl} = T + T_k$ is the ZP-OFDM data block duration, and $g(t)$ describes the zero-padding operation, i.e.

$$g(t) = \begin{cases} 
1, & t \in [0, T] \\
0, & \text{elsewhere.} 
\end{cases} \tag{4}$$

In both CP-OFDM and ZP-OFDM, transmitted symbols are divided into 3 groups: data symbols $S_D$, pilot symbols $S_P$ and null symbols $S_N$. The main purpose of pilot and null subcarriers is to facilitate channel estimation and residual Doppler shift mitigation, respectively [56]. They can also be used for Doppler scale estimation [97], which will be discussed in Chapter 3.

### 2.2 Channel Model

In this thesis, the considered multipath channel consists of $N_{pa}$ paths as:

$$h(t; \tau) = \sum_{p=1}^{N_{pa}} A_p(t) \delta(t - \tau_p(t)), \tag{5}$$

where $A_p(t)$ and $\tau_p(t)$ denote the amplitude and delay of the $p$-th path, respectively. According to the study of [107], throughout this thesis, we assume that the amplitude is constant within each OFDM block (about 200 ms), i.e., $A_p(t) \approx A_p$, which leads to

$$h(t; \tau) = \sum_{p=1}^{N_p} A_p \delta(t - \tau_p(t)). \tag{6}$$

Following two assumptions can be made on the delay $\tau_p(t)$
(i) Common Doppler scale factor \( a \) on all the paths:

\[
\tau_p(t) = \tau_p - at. \tag{7}
\]

(ii) Different Doppler scale factors \( a_p \) on different paths:

\[
\tau_p(t) = \tau_p - a_p t. \tag{8}
\]

So the simplified channel models are:

(i) \[
h(t; \tau) = \sum_{p=1}^{N_{pa}} A_p \delta(t - [\tau_p - at]), \tag{9}\]

(ii) \[
h(t; \tau) = \sum_{p=1}^{N_{pa}} A_p \delta(t - [\tau_p - a_p t]), \tag{10}\]

2.3 Receiver Process

2.3.1 Time Domain Received Signal

Assume that the receiver has \( N_r \) hydrophones. The continuous time input-output relationship on the \( \nu \)-th hydrophone can be expressed as:

\[
y_\nu(t) = h_\nu(t) \ast x(t) + w_\nu(t), \tag{11}\]

where \( y_\nu(t) \) is the received sample at the \( \nu \)-th hydrophone; \( h_\nu(t) \) is the channel impulse response at the \( \nu \)-th hydrophone; \( x(t) \) is the transmitted signal; and \( w_\nu(t) \) is the additive ambient noise at the \( \nu \)-th hydrophone.

The discrete time model for Equation (11) is:

\[
y_\nu[n] = h_\nu[n] \ast x[n] + w_\nu[n], \tag{12}\]
2.3.2 Frequency Domain Signal

As described in [56], a two-step approach will be applied to the received time domain signal $y_\nu[n]$ in (12) to mitigate the Doppler effect. First, resampling operation will be carried out to compensate the main Doppler offset, which converts a “wideband” problem into a “narrowband” problem. Then, high-resolution uniform compensation will be carried out to deal with the residual Doppler, which corresponds to the “narrowband” model for best ICI reduction.

After the two-step Doppler mitigation, time domain signal is then converted into frequency domain through overlap-add fast Fourier transform (FFT) process. We have the equivalent frequency domain input-output relationship as:

$$z_\nu[k] = H_\nu[k]s[k] + \xi_\nu[k],$$  \hspace{1cm} (13)

where $z_\nu[k]$ is the observation at the $k$-th subcarrier of the $\nu$-th hydrophone; $H_\nu[k]$ is the channel transfer function of the $k$-th subcarrier on the $\nu$-th hydrophone; $s[k]$ is the transmitted symbol on the $k$-th subcarrier; and $\xi_\nu[k]$ is the sum of additive ambient noise and inter-carrier interference (ICI) at the $k$-th subcarrier of the $\nu$-th hydrophone.

If we replace $H_\nu[k]$ with its estimate value $\hat{H}_\nu[k]$ in Equation (13), we will have:

$$z_\nu[k] = \hat{H}_\nu[k]s[k] + \eta_\nu[k],$$  \hspace{1cm} (14)
where $\eta_\nu[k]$ now includes ambient noise, residual ICI and channel estimation error.

We assume that noise at all receive elements are independently and identically distributed (i.i.d.), with $\eta_\nu[k] \sim \mathcal{CN}(0, \sigma_\eta^2)$, $\forall \nu$.

### 2.3.3 Maximum-Ratio Combining (MRC)

Applying MRC to Equation (14):

$$z_{\text{mrc}}[k] = \frac{\sum_{\nu=1}^{N_r} \hat{H}^*_\nu[k] z_\nu[k]}{\sqrt{\sum_{\nu=1}^{N_r} \hat{H}^*_\nu[k] \hat{H}_\nu[k]}}.$$  \hspace{1cm} (15)

Define $\hat{H}_{\text{mrc}}[k]$ as the equivalent channel gain

$$\hat{H}_{\text{mrc}}[k] = \sqrt{\sum_{\nu=1}^{N_r} \hat{H}^*_\nu[k] \hat{H}_\nu[k]}.$$  \hspace{1cm} (16)

The equivalent channel input-output relationship is

$$z_{\text{mrc}}[k] = \hat{H}_{\text{mrc}}[k] s[k] + \eta_{\text{mrc}}[k],$$  \hspace{1cm} (17)

where the equivalent noise $\eta_{\text{mrc}}[k]$ contains ambient noise, residual ICI and channel estimation error. It can be shown that equivalent noise is

$$\eta_{\text{mrc}}[k] = \frac{1}{\hat{H}_{\text{mrc}}[k]} \sum_{\nu=1}^{N_r} \hat{H}^*_\nu[k] \eta_\nu[k],$$  \hspace{1cm} (18)

which can be verified to still follow complex Gaussian distribution $\eta_{\text{mrc}}[k] \sim \mathcal{CN}(0, \sigma_\eta^2)$. 

2.3.4 Log Likelihood Ratio (LLR) Computation

Based the input-output relationship after MRC (17), the least squares (LS) estimate of the information symbol is

$$\hat{s}_{LS}[k] = \frac{\hat{z}_{\text{mrc}}[k]}{H_{\text{mrc}}[k]}.$$  \tag{19}

Assume transmitted symbol $s[k]$ is taken from a size-$M$ constellation $\{\alpha_0, ..., \alpha_{M-1}\}$, then LLR for the $i$-th symbol $\alpha_i$ is calculated as [119, Chapter 8]:

$$L_i[k] = 2 \frac{\Re\{z_{\text{mrc}}^*[k]H_{\text{mrc}}[k](\alpha_i - \alpha_0)\}}{\sigma_n^2} - \frac{1}{\sigma_n^2} |H_{\text{mrc}}[k]|^2(|\alpha_i|^2 - |\alpha_0|^2).$$  \tag{20}

The calculated LLR will be the input to a channel decoder.

2.4 Connection between This Chapter and Following Chapters

Section 3.1 introduces the work of Doppler scale estimation for underwater acoustic OFDM. All of the Doppler scale estimation methods are based on signal structure introduced in Section 2.1. Channel models introduced in Section 2.2 are used in simulations to verify the system performance of different Doppler scale estimation methods.

Section 3.2 introduces AMC for underwater acoustic OFDM. Similar signal structure as shown in Section 2.1 has been adopted for this AMC OFDM system. Channel models in Section 2.2 has been applied to test the BLER performance of different transmission modes. The previous and proposed performance metrics for AMC operation are all based on receiver process introduced in Section 2.3.1, 2.3.2 and 2.3.3.
Section 4.1 describes the work of SISO and MIMO OFDM DSP optimization for underwater acoustic modems. The signal structure and receiver process are the same as those introduced in Sections 2.1 and 2.3.

Section 4.2 is on implementation of OFDM-DCC in underwater acoustic channels. The major work in this part of thesis is on top of Section 4.1, which adopts the system structure described in Sections 2.1 and 2.3.

Chapter 5 presents the work of field performance analysis of underwater acoustic OFDM. Signal structure adopted in the deployment is similar as that described in Section 2.1. Performance metrics calculated based on the described signal in Section 2.3.1, 2.3.2 are included to measure system performance. Data-driven sparsity factor optimization and effective noise based multichannel combining techniques are developed directly from (14) in Section 2.3.2 to improve the data decoding performance.
Chapter 3

Algorithm Design for Underwater Acoustic OFDM

3.1 Comparison of Doppler Scale Estimation Methods for Underwater Acoustic OFDM

3.1.1 Introduction to Doppler Scale Estimation

As mentioned in Section 1.2, Doppler scale estimation is a critical step in underwater acoustic OFDM receivers. Only with accurately estimated Doppler scale factors, frequency offset compensation can be correctly carried out to restore the orthogonality among different subcarriers in an OFDM signal. In previous works like [74, 56, 116], special attention has been paid to Doppler scale estimation.

Typically, Doppler scale estimation is accomplished by inserting waveforms known to the receiver during the data transmission. Two popular approaches are in the following.
(i) One approach is to use a pulse train which is formed by the repetition of a *Doppler-insensitive waveform* [50], such as linear-frequency modulated (LFM) waveform [51] and hyperbolic-frequency modulated (HFM) waveform [52]. A transmission format with one preamble and one postamble around the data burst is usually adopted [115, 74, 56], as shown in Fig. 2. At the receiver side, by detecting the times-of-arrival of the preamble and postamble, thus the interval change in-between, an average Doppler scale estimate over the whole data burst can be obtained. Thanks to the Doppler-insensitive property of the waveforms, a single-branch-matched filtering operation is adequate even in the presence of Doppler distortion. However, this method is only suitable for offline processing due to the processing delay.

(ii) The other approach is to use a *Doppler-sensitive waveform with a thumbtack ambiguity function*. A Doppler-sensitive waveform is usually transmitted as a preamble prior to the data burst, as shown in Fig. 2. At the receiver side, a bank of correlators correlate the received signal with preambles pre-scaled by different Doppler scaling factors, and the branch with the largest correlation peak provides the estimated Doppler scale [115]. Typical Doppler-sensitive waveforms include Costa waveforms [24], m-sequence [70], and poly-phase sequence [31].
In this thesis, we focus on an underwater acoustic communication system that adopts the signal structure as shown in Fig. 1. By exploiting the cyclic repetition structure of the CP-OFDM preamble, a blind estimation with a bank of self-correlators was proposed in [65]. However, it does not leverage the knowledge of the waveform itself which is known to the receiver. Taking this method as the first approach, one can easily construct the following Doppler scale estimators for the OFDM transmission in Fig. 1.

(i) *Cross-correlation with the CP-OFDM preamble:* Given the Doppler sensitivity of the OFDM waveform, a bank of cross-correlators can use the pre-scaled versions of the CP-OFDM waveform as local replicas.

(ii) *Pilot-aided method for each ZP-OFDM block:* By taking the waveform constituted by the pilot-subcarrier components as a replica of the transmitted signal, the Doppler estimation method using a bank of cross-correlators is directly applicable.
(iii) **Null-subcarrier based blind estimation method for each ZP-OFDM block:** As an extension of the blind carrier frequency offset (CFO) estimation method [63], the receiver rescales the received waveform with different tentative Doppler scaling factors, and uses the energy on the null subcarriers to find the best fit.

(iv) **Decision-aided method for each ZP-OFDM block:** Once a ZP-OFDM block is successfully decoded, the transmitted waveform corresponding to this block can be reconstructed at the receiver. Taking the reconstructed waveform as a local replica, the Doppler estimation method using a bank of correlators can be deployed to refine the Doppler scale estimation for this block. The refined Doppler scale estimate can be passed to the next block.

### 3.1.2 Doppler Scale Estimation with A CP-OFDM Preamble

Consider a CP-OFDM preamble structure in Fig. 1, which consists of two identical OFDM symbols of length $T_0$ and a cyclic prefix of length $T_{cp}$ in front, with the embedded structure:

$$x_{cp}(t) = x_{cp}(t + T_0), \quad -T_{cp} \leq t \leq T_0. \tag{21}$$

Let $B$ denote the system bandwidth, and define $K_0 := BT_0$ as the number of subcarriers, baseband CP-OFDM signal can be expressed in (1). The passband signal can be obtained as $\tilde{x}_{cp}(t) = 2\text{Re}\{x_{cp}(t)e^{j2\pi f_c t}\}$, where $f_c$ is the center frequency.
After transmitting the passband signal $\tilde{x}_{cp}(t)$ through the multipath channel shown in Equation (5), the received passband signal $\tilde{y}(t)$ is converted to baseband as $y(t) = \text{LPF}(\tilde{y}(t)e^{-j2\pi f_c t})$, where LPF denotes the low pass filtering operation.

### 3.1.2.1 Self-Correlation

If all the paths in the channel have the same Doppler scale factor $a$, then the multipath channel can be simplified to be Equation (9). It is shown in [65] that the embedded structure in the received waveform becomes

$$y(t) = e^{-j2\pi \frac{a}{1+a} f_c T_0} y(t + \frac{T_0}{1+a}), \quad -\frac{T_{cp} - \tau_{\text{max}}}{1+a} \leq t \leq \frac{T_0}{1+a}, \quad (22)$$

which has a repetition period $T_0/(1+a)$ regardless of the channel amplitudes.

By exploiting the structure in (22), the time-of-arrival and the Doppler scale of the CP-OFDM symbol in the received signal can be jointly estimated via

$$\left(\hat{a}, \hat{\tau}\right) = \arg\max_{a, \tau} \left| \int_0^{\frac{T_0}{1+a}} y(t + \tau) y^* \left(t + \tau + \frac{T_0}{1+a}\right) dt \right|, \quad (23)$$

which does not require the knowledge of the channel and the data symbols. This method can be implemented with a bank of self-correlators [65].

### 3.1.2.2 Cross-Correlation

Rather than exploiting the structure of the CP-OFDM preamble, the cross-correlation based method can be used, since the transmitted preamble is known at the receiver. Taking the basic unit of duration $T_0$ as the template, the joint
time-of-arrival and Doppler rate estimation can be achieved via
\[
(\hat{a}, \hat{\tau}) = \arg\max_{a,\tau} \left| \int_0^{T_d} y(t + \tau)x^*_c ((1 + a)t) e^{-j2\pi af_c t} dt \right|.
\] (24)

This can be implemented via a bank of cross-correlators, where the branch yielding the largest peak provides the needed Doppler scale estimate.

### 3.1.3 Doppler Scale Estimation with Each ZP-OFDM Block

As described in Chapter 2, each ZP-OFDM symbol consists of the multiplexing of data symbols \( S_D \), pilot symbols \( S_P \) and null \( S_N \). Denote \( K \) to be the number of subcarriers, \( T \) to be symbol duration, \( T_g \) to be guard interval, and \( T_{bl} := T + T_g \) to be the total ZP-OFDM block duration. Then we have \( S_D \cup S_P \cup S_N = \{-K/2, ..., K/2 - 1\} \). The baseband transmitted ZP-OFDM signal of Equation (3) can also be expressed by
\[
x_{zp}(t) = \sum_{k \in S_D \cup S_P} s[k]e^{j2\pi \frac{k}{K} t} g(t), \quad t \in [0, T_{bl}]
\] (25)

where \( g(t) \) still describes the zero-padding operation.

After transmitting the ZP-OFDM symbol through a multipath channel defined in (6), we denote \( \tilde{y}(t) \) as the received passband signal, whose baseband version is \( y(t) = \text{LPF}(\tilde{y}(t)e^{-j2\pi f_c t}) \). The availability of null subcarriers, pilot subcarriers, and data subcarriers can be used for Doppler scale estimation.
3.1.3.1 Null-Subcarrier Based Blind Estimation

In [65], the null subcarriers in ZP-OFDM system are exploited to perform carrier frequency offset (CFO) estimation. Here in this thesis, the same principle is used to estimate Doppler scale factor.

Assume that coarse synchronization is available from the preamble. After truncating each ZP-OFDM block from the received signal, we resample one block with different tentative scaling factors. The total energy of frequency measurements at null subcarriers are used as a metric for the Doppler scale estimation

\[
\hat{a} = \arg \min_a \sum_{k \in S_N} \left\| \int_0^{T+T_k} y\left(\frac{t}{1+a}\right) e^{-j2\pi a f_c t} e^{-j2\pi a f_c t} dt \right\|^2.
\]  

(26)

For each tentative \(a\), a resampling operation is carried out followed by fast Fourier transform. A one-dimensional grid-search leads to a Doppler scale estimate.

3.1.3.2 Pilot-Aided Estimation

As introduced above, a set of subcarriers \(S_p\) is dedicated to transmit pilot symbols. Hence, the transmitted waveform \(x_{zp}(t)\) is partially known, containing

\[
x_{\text{pilot}}(t) = \sum_{k \in S_p} s[k] e^{j2\pi k t} g(t), \quad t \in [0, T].
\]  

(27)

The joint time-of-arrival and Doppler scale estimation is achieved via

\[
(\hat{a}, \hat{\tau}) = \arg \max_{a, \tau} \left| \int_0^{T} y(t + \tau) x_{\text{pilot}}^* ((1 + a)t - \tau) e^{-j2\pi a f_c t} dt \right|.
\]  

(28)

which can be implemented via a bank of cross-correlators.
3.1.3.3 Decision-Aided Estimation

For an OFDM transmission with multiple blocks, the Doppler estimated in one block can be used for the resampling operation of the next block assuming small Doppler variation across blocks. After the decoding operation the receiver can reconstruct the transmitted time-domain waveform, by replacing $s[k]$ by its estimate $\hat{s}[k]$, $\forall k \in S_D$ in Equation (25). Denote the reconstructed waveform as $\hat{x}_{zp}(t)$.

Similar to the pilot-aided method, the decision-aided method performs the joint time-of-arrival and Doppler scale estimation via

$$
(\hat{a}, \hat{\tau}) = \arg \max_{a,\tau} \left| \int_0^{T_r} y(t + \tau)\hat{x}_{zp}^* ((1 + a)t - \tau) e^{-j2\pi a f_c t} dt \right|
$$

which again, is implemented via a bank of cross-correlators. The estimated $\hat{a}$ can be used for the resampling operation of the next block.

**Remark 1:** Relative to the pilot-aided method, the decision-aided method leverages the estimated information symbols, thus is expected to achieve a better estimation performance. Assuming that all the information symbols have been successfully decoded, the decision-aided method has knowledge about both the data and pilot symbols. Let $|S_P|$ and $|S_D|$ denote the numbers of pilot and data symbols, respectively. Using the template $\hat{x}_{zp}(t)$ constructed from $(|S_P| + |S_D|)$ known symbols for cross correlation achieves a $10 \log_{10}((|S_P| + |S_D|)/|S_P|)$ dB power gain in terms of noise reduction, relative to that using the template $x_{\text{pilot}}(t)$ constructed from $|S_P|$ known symbols.
3.1.4 Simulation Results

The OFDM parameters are summarized in Table 1. For CP-OFDM, the data symbols at all the 512 subcarriers are randomly drawn from a quadrature phase-shift keying (QPSK) constellation. For ZP-OFDM, out of 1024 subcarriers, there are $|S_N| = 96$ null subcarriers with 24 on each edge of the signal band for band protection and 48 evenly distributed in the middle for the carrier frequency offset estimation; $|S_P| = 256$ are pilot subcarriers uniformly distributed among the 1024 subcarriers, and the remaining are $|S_D| = 672$ data subcarriers for delivering information symbols. The pilot symbols are drawn randomly from a QPSK constellation. The data symbols are encoded with a rate-$1/2$ nonbinary low-density parity-check (LDPC) code [43] and modulated by a QPSK constellation.

Table 1: OFDM Parameters in Simulations

<table>
<thead>
<tr>
<th>System Parameters:</th>
<th>CP-OFDM</th>
<th>ZP-OFDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center frequency: $f_c$</td>
<td>13 kHz</td>
<td>13 kHz</td>
</tr>
<tr>
<td>Bandwidth: $B$</td>
<td>4.88 kHz</td>
<td>4.88 kHz</td>
</tr>
<tr>
<td># of subcarriers: $K_0 = 512$</td>
<td></td>
<td>$K = 1024$</td>
</tr>
<tr>
<td>Time duration: $T_0 = 104.86$ ms</td>
<td></td>
<td>$T = 209.72$ ms</td>
</tr>
<tr>
<td>Guard interval: $T_{cp} = 100$ ms</td>
<td></td>
<td>$T_k = 40.3$ ms</td>
</tr>
</tbody>
</table>

Three UWA channel settings are tested, which are Equation (5) with different parameters.

- **Channel Setting 1**: A single-path channel. In this case, Equation (5) is simplified to

$$h(t, \tau) = \delta(t - [\tau - at]). \quad (30)$$
• **Channel Setting 2:** A 15 path channel, where all paths have one common Doppler scaling factor, which is indeed Equation (9) with \( N_{pa} = 15 \).

• **Channel Setting 3:** A 15 path channel, where each path has an individual Doppler scaling factor, which is Equation (10) with \( N_{pa} = 15 \).

The inter-arrival-time of paths follows an exponential distribution with a mean of 1 ms. The mean delay spread for the channels in channel setting 2 and 3 is thus 15 ms. The amplitudes of paths are Rayleigh distributed with the average power decreasing exponentially with the delay, where the difference between the beginning and the end of the guard time is 20 dB. For each path, the Doppler scale \( a_p \) is generated from a Doppler speed \( v_p \) (with unit m/s)

\[
a_p = v_p / c,
\]

where \( c = 1500 \text{ m/s} \) is the sound speed in water. In channel settings 1 and 2, the Doppler speed \( v \) is uniformly distributed within \([-4.5, 4.5]\) m/s. In channel setting 3, the Doppler speeds \( \{v_p\} \) are randomly drawn from the interval \([1.5 - 0.1, 1.5 + 0.1]\) m/s.

In channel settings 1 and 2, the ground truths of \( v \) and \( a \) are known. We adopt the root-mean-squared-error (RMSE) of the estimated Doppler speed as the performance metric,

\[
RMSE = \sqrt{E[|\hat{v} - v|^2]} = \sqrt{E[(\hat{a} - a)c^2]} \tag{32}
\]

which has the unit m/s. In channel setting 3, different paths have different Doppler scales, while the Doppler scale estimator only provides one estimate.
RMSE is hence not well motivated. With the estimated Doppler scale to perform the resampling operation, we will use the block-error-rate (BLER) of the ZP-OFDM decoding as the performance metric.

3.1.4.1 RMSE Performance with CP-OFDM

For the single-path channel, Fig. 3 shows the RMSE performance of two estimation methods at different signal-to-noise ratio (SNR) levels. One can see a considerable gap between the self-correlation method and the cross-correlation method, while in the medium to high SNR region, both methods can provide a reasonable performance to facilitate receiver decoding.

![Graph showing RMSE performance with CP-OFDM](image)

Figure 3: Performance of different estimators for the CP-OFDM preamble in single-path and multipath channels (channel settings 1 and 2).

For the multipath channel with a single Doppler speed, Fig. 3 shows the RMSE performance of two estimation methods. One can see that the cross-correlation method outperforms the self-correlation method considerably in the low SNR
region. However, the former suffers an error floor in the high SNR region, while the later does not.

Relative to the RMSE performance in the single-path channel, a considerable performance degradation can be observed for the cross-correlation method in the multipath channel, whereas the performance of the self-correlation method is quite robust. The reason for the difference lies in the capability of the self-correlation method to collect the energy from all paths for Doppler scale estimation, while the cross-correlation method aims to get the Doppler scale estimate from only one path, the strongest path.

3.1.4.2 RMSE Performance with ZP-OFDM

Fig. 4 shows the RMSE performance of three estimation methods for ZP-OFDM in single-path channels. In the low SNR region, one can see that the decision-aided method is the best, while the null-subcarrier based blind method is the worst. As discussed in Remark 1, the decision-aided method achieves $10 \log_{10} \left( \frac{|S_D| + |S_P|}{|S_P|} \right) \approx 6$ dB power gain relative to the pilot-aided method. In the medium and high SNR region, the pilot-aided method suffers an error floor due to the interference from the data subcarriers, and the null-subcarrier based blind method gets a good estimation performance. The Cramer-Rao lower bound (CRLB) with a known waveform is also included as the performance benchmark, whose derivation can be carried out similar to [34, 67].
Figure 4: Performance of different estimators for ZP-OFDM in single-path channels (channel setting 1). The CRLB with all data known is included as a benchmark.

Fig. 5 shows the RMSE performance of three methods in multipath channels with a common Doppler speed. For each realization, the Doppler scale, the path amplitudes and delays are randomly generated. The RMSE corresponding to each method is calculated by averaging the estimation error over multiple realizations. Again, one can see that in the low SNR region, the decision-aided method has the best performance, while the null-subcarrier based blind method is the worst. Different from the performance in the single-path channel, the decision-aided method has an error floor in the high SNR region, since it only picks up the maximum correlation peak of one path. On the other hand, the null-subcarrier method has robust performance in the presence of multiple paths.

3.1.4.3 Comparison of Blind Methods of CP- and ZP-OFDM

The self-correlation method for the CP-OFDM preamble is closely related to the null-subcarrier based blind method for ZP-OFDM. This can be easily verified
Figure 5: Performance of different estimators for ZP-OFDM in multipath channels with a common Doppler scale (channel setting 2).

by rewriting (1) as

$$x_{cp}(t) = \sum_{k=-K_0}^{K_0-1} s'[k] e^{j2\pi \frac{2\pi k}{2T_0} t} q(t), \quad t \in [-T_{cp}, 2T_0]$$

where $s'[k] = 0$ when $k$ is odd and $s'[k] = s[k/2]$ when $k$ is even. The cyclic repetition pattern in (21) is generated by placing zeros on all odd subcarriers in a long OFDM symbol of duration $2T_0$. Hence, the self-correlation implementation could be replaced by the null-subcarrier based implementation for the CP-OFDM preamble.

Fig. 6 shows the performance comparison between the blind method for ZP-OFDM and that for CP-OFDM in the multipath channel with one Doppler scale factor, respectively. At low SNR, typically when it’s lower than 0 dB, the null-subcarrier based method in CP-OFDM system has a better performance than that in the ZP-OFDM system, which is due to the fact that CP-OFDM system has 512 null subcarriers, more than 96 null subcarriers in the ZP-OFDM block. At high
SNR, the null subcarrier based method in ZP-OFDM has better performance. The possible reason is that null subcarriers in ZP-OFDM are distributed with an irregular pattern, which could outperform the regular pattern in the CP-OFDM preamble.

![Graph showing RMSE of Doppler speed vs. Input SNR for ZP and CP OFDM methods](image)

**Figure 6:** Null subcarrier based method in ZP-OFDM and CP-OFDM.

### 3.1.4.4 BLER Performance with ZP-OFDM

With channels generated according to the channel setting 3, Fig. 7 shows the simulated BLER performance, where the received OFDM blocks are resampled with the Doppler scale estimates from different estimators and processed using the receiver from [56] and the LDPC decoder from [43]. At each SNR point, at least 20 block errors are collected.

It is expected that the OFDM system can only work when the useful signal power is above that of the ambient noise. Regarding the simulation results in Fig. 5, one can see that all the methods can reach a RMSE lower than 0.1 m/s.
Hence, it is not surprising that these methods lead to quite similar BLER results as shown in Fig. 7. This observation is consistent with the analysis in [65] that an estimation error of 0.1 m/s introduces a negligible error.

![Figure 7: The BLER performance in multipath multi-Doppler channels (channel setting 3).](image)

### 3.1.5 Experimental Results

This *mobile acoustic communication experiment* (MACE10) was carried out off the coast of Martha’s Vineyard, Massachusetts, June, 2010. The water depth was about 80 meters. The receiving array was stationary, while the source was towed slowly away from the receiver and then towed back, at a speed around 1 m/s. The relative distance of the transmitter and the receiver changed from 500 m to 4.5 km. Out of the two tows in this experiment, we only consider the data collected in the first tow. There are 31 transmissions in total, with a CP-OFDM preamble and 20 ZP-OFDM blocks in each transmission. We exclude one
transmission file recorded during the turn of the source, where the SNR of the received signal is quite low.

The CP-OFDM and ZP-OFDM parameters and signal structures are identical to that in the simulation, as listed in Table 1.

![Figure 8: MACE10: Estimated Doppler speeds for 30 data bursts in MACE10, where each data burst has 20 OFDM blocks. The time interval between two consecutive date bursts is around 4 mins.](image)

![Figure 9: Estimated channel impulse responses for two different blocks at different bursts.](image)
Fig. 8 shows the estimated Doppler speeds for ZP-OFDM blocks from different methods. Clearly, the Doppler speed fluctuates from block to block. Fig. 9 shows the estimated channel impulse responses for two ZP-OFDM blocks from two data sets, where the time interval between these two data bursts is more than 1 hour. The channels have a delay spread about 20 ms but with different delay profiles.

Based on the recorded files, we carried out two tests.

3.1.5.1 Test Case 1

In this test, we focus on one single file (file ID: 1750155F1954_C0_S5), and compare the RMSE performance of different approaches by adding artificial noise to the recorded signal. The ground truth of the Doppler scale factor is not available. When computing the RMSE using (32) for each method, we use the estimated Doppler scale of the original file without adding the noise as the ground truth. Fig. 10 shows the estimation performance of several approaches. Similar observations as the simulation results in Figs. 3 and 5 are found.

3.1.5.2 Test Case 2

In this test, we compare the BLER performance of an OFDM receiver where the resampling operation is carried out with different Doppler scale estimates from different methods.

Due to the relatively high SNR of the recorded signal, we create a semi-experimental data set by adding white Gaussian noise to the received signal.
Figure 10: MACE10: Performance comparison of Doppler estimation approaches, file ID: 1750155F1954_C0_S5.

Define $\hat{\sigma}^2$ as the estimated ambient noise power in the original recorded signal. Fig. 11 shows the BLER performance with different Doppler estimation approaches by adding different amount of noises to the received files.

Figure 11: MACE10: BLER Performance using different Doppler estimation methods by adding artificial noise to the received signal, $\hat{\sigma}^2$ denoting the estimated ambient noise power.
One can see that the methods for ZP-OFDM outperforms the methods for CP-OFDM, as the Doppler scale itself is continuously changing from block to block, as illustrated in Fig. 8.

Another interesting observation is that the null-subcarrier based blind method has slight performance improvement relative to the pilot- and decision-aided methods. This agrees with the simulation results in Fig. 5 that in the high SNR region, the blind estimation method does not suffer an error floor in the multipath channel, hence enjoys a better estimation performance.

3.1.6 Extension to Distributed MIMO-OFDM

If the transmitters in a multi-input multi-output (MIMO) system are co-located, the Doppler scales corresponding to all transmitters are similar, and hence a single-user blind Doppler scale estimation method would work well, as done in [55]. However, if the transmitters are distributed, for example in a system with multiple single-transmitter users, the Doppler scales for different users could be quite different, even with opposite signs [92]. We now investigate the performance of different Doppler scale estimation methods in the presence of multiuser interference. We will use the ZP-OFDM waveform as the reference design; similar conclusions can be applied to the CP-OFDM preamble. Only simulated data sets are used in the following tests.
3.1.6.1 Pilot- and Decision-aided Estimation

We simulate a two-user system. Each user generates a multipath channel according to channel setting 2 independently. The positions of pilot, null, and data subcarriers are the same for different users. The pilot and data symbols of different users are randomly generated and hence are different.

Fig. 12 depicts the RMSE performance of the pilot- and decision-aided estimation methods. Compared with the performance in the single-user setting in Fig. 5, there are performance degradation and the error floors are higher. However, both methods can achieve RMSE lower than 0.1 m/s at low SNR values. Hence, both methods have robust performance in the presence of multiuser interference.
3.1.6.2 Null-Subcarrier Based Blind Estimation

The null-subcarrier based blind estimation method exploits the transmitted OFDM signal structure. Since all the users share the same positions of null subcarriers, there is a user-association problem even when multiple local minimums are found. We simulate a two-user system where the Doppler speeds of user 1 and user 2 are uniformly distributed within $[-4.5, -0.5]$ m/s and $[0.5, 4.5]$ m/s, respectively. Without adding the ambient noise to the received signal, Fig. 13 demonstrates both successful and failed cases using the objective function in (26). The objective functions in the single-user settings are also included for comparison. One can see that the multiuser interference degrades the estimation performance significantly. Hence, although the blind method developed for the single user case can be used to co-located MIMO-OFDM as in [55], it is not applicable to distributed MIMO-OFDM where different users have different Doppler scales.

3.1.7 Summary

This section of thesis compared different methods for Doppler scale estimation for a CP-OFDM preamble followed by ZP-OFDM data transmissions. Blind methods utilizing the underlying signalling structure work very well at medium to high SNR ranges, while cross-correlation based methods can work at low SNR ranges based on the full or partial knowledge of the transmitted waveform. All
of these methods are viable choices for practical OFDM receivers. In a distributed multiuser scenario, cross-correlation approaches are more robust against multiuser interference than blind methods.

3.2 Adaptive Modulation and Coding (AMC) for Underwater Acoustic OFDM

3.2.1 Introduction to AMC for Underwater Acoustic OFDM

As described in Section 1.3, AMC technique is appealing for underwater acoustic communications to improve the system efficiency. The study on the application of AMC to underwater acoustic communications has been limited in comparison to the extensive investigations on receiver designs with fixed modulations. In [64], a single carrier PSK based AMC system is proposed, in which
both the constellation size and turbo code rate are adjustable. The index for different working modes is achievable information rate with i.i.d. Gaussian inputs and post-equalization SNR. In [90], a Nakagami-\(m\) based channel model has been proposed to simulate the channel behavior of real data sets, which is then used to predict the performance of adaptive modulation based on symbol SNR. Recently, an adaptive OFDM system that maximizes throughput under the constraint of certain target bit-error-rate (BER) is proposed in [71], where the system uses the predicted channel to decide the optimal modulation sizes and power levels for different OFDM subcarriers. Sea trial results are also presented, with feedback implemented from a radio frequency (RF) link [71].

In this section of thesis, we study adaptive modulation and coding for underwater acoustic OFDM based on a finite number of transmission modes. In first part of this study, the objective is to maximize the data rate via mode switching. In the second part of this study, we explore the “green communications” concept, a popular topic in radio communications [27, 21, 11, 38], in the context of underwater acoustic communications by minimizing the energy consumption for a finite-length data packet through mode switching. Power measurements from the OFDM modem platform [3] are used in the example study for energy minimization.
3.2.2 Construction of Transmission Modes

The first task of the AMC-OFDM design is to prepare a set of transmission modes for underwater OFDM with different data rates and performance.

3.2.2.1 Modulation and Coding Pairs

Nonbinary low-density parity-check (LDPC) coded modulation has been adopted in [44] for underwater OFDM, where the size of the Galois field matches the constellation size. A set of modulation and coding pairs have been provided, out of which the rate 1/2, length 672 LDPC code in Galois field(4) (GF(4)) in combination with QPSK modulation has been implemented in an OFDM modem prototype [111].

Due to the implementation complexity of high order Galois field, we pursue LDPC coding in GF(4), and suitably match the coded symbols to BPSK, QPSK, and 16 quadrature amplitude modulation (16QAM) constellations. Table 2 lists the five transmission modes, where $r_c$ denotes the code rate. While modes 1 and 2 are identical to those used in [44], modes 3, 4, and 5 are newly constructed. For mode 1, every coded symbol in GF(4) is mapped to two BPSK symbols. For modes 2 and 3, every coded symbol in GF(4) is mapped to one QPSK symbol. For modes 4 and 5, every two coded symbols in GF(4) are mapped to one 16QAM symbol, one to the real part and the other to the imaginary part. Hence, all the five modulation and coding pairs in Table 2 lead to 672 coded symbols after coding and modulation.
Table 2: Modulation and coding pairs for AMC OFDM

<table>
<thead>
<tr>
<th>Index</th>
<th>Parity check matrix size</th>
<th>Code rate $r_c$</th>
<th>Modulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$168 \times 336$</td>
<td>$1/2$</td>
<td>BPSK</td>
</tr>
<tr>
<td>2</td>
<td>$336 \times 672$</td>
<td>$1/2$</td>
<td>QPSK</td>
</tr>
<tr>
<td>3</td>
<td>$168 \times 672$</td>
<td>$3/4$</td>
<td>QPSK</td>
</tr>
<tr>
<td>4</td>
<td>$672 \times 1344$</td>
<td>$1/2$</td>
<td>16QAM</td>
</tr>
<tr>
<td>5</td>
<td>$336 \times 1344$</td>
<td>$3/4$</td>
<td>16QAM</td>
</tr>
</tbody>
</table>

Fig. 14 shows the block error rate (BLER) performance for the 5 modes in Table 2, with both additive white Gaussian noise (AWGN) channel and i.i.d. Rayleigh fading channel with four receive elements for diversity combining. Solid lines with circle markers correspond to the simulated performance of the 5 modes listed in Table 2, with perfect channel knowledge. About 3 dB difference can be observed between consecutive modes in the i.i.d. channel with 4 receivers. Dashed bold lines correspond to the capacity limit in the AWGN channel and the outage probability in the Rayleigh fading channel when the source symbols can be assumed to be Gaussian distributed. The solid bold lines correspond to the capacity limit in the AWGN channel and the outage probability in the Rayleigh fading channel when the source symbols are drawn from the same constellations as the transmission modes in Table 2; please see e.g., [93, 62, 29] on how to compute the information-theoretic limits with a finite-alphabet input.

With finite block lengths, the transmission modes constructed in Table 2 are within $2 \sim 3$ dB away from the information-theoretic limits that assume infinite code length and optimal decoding. This demonstrates that the designed modulation and coding pairs have satisfactory performance.
3.2.2.2 Transmission Modes on an OFDM Modem Platform

Now we incorporate the modulation and coding pairs in Table 2 to the zero-padded OFDM transmission as implemented in [111]. The OFDM bandwidth is $B = 6000$ Hz and the total number of subcarriers is $K = 1024$, which leads to a symbol duration $T = 170.7$ ms. Out of $K = 1024$ subcarriers, there are $|S_P| = 256$ pilot subcarriers, $|S_N| = 96$ null subcarriers and $|S_D| = 672$ data subcarriers. Hence each codeword can be accommodated in one OFDM symbol, no matter which mode is used.

Use $M$ as the modulation size, and $T_g$ as the guard zero length in zero padded OFDM. The data rate $R$ for all the modes can be calculated as:

$$R = \frac{r_c |S_D| \log_2 M}{T + T_g}$$ (34)
The guard interval $T_g$ can be varied easily by the OFDM transmitter. With $T_g = 50$ ms, the corresponding data rates for the five transmission modes are listed in Table 3.

**Table 3**: Payload of the 5 modes with $T_g = 50$ ms (Note that 4 bytes reserved by the modem physical layer are excluded in the computation)

<table>
<thead>
<tr>
<th>Mode</th>
<th>Payload per block (bytes)</th>
<th>$R$ (kb/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td>38</td>
<td>1.38</td>
</tr>
<tr>
<td>Mode 2</td>
<td>80</td>
<td>2.90</td>
</tr>
<tr>
<td>Mode 3</td>
<td>122</td>
<td>4.42</td>
</tr>
<tr>
<td>Mode 4</td>
<td>164</td>
<td>5.94</td>
</tr>
<tr>
<td>Mode 5</td>
<td>248</td>
<td>8.99</td>
</tr>
</tbody>
</table>

Figure 15: The data transmission format.

We implement these five transmission modes on the OFDM modem hardware platform [3], where the transceiver algorithms run on the TMS320C6747 DSP core from Texas Instruments. As shown in Fig. 15, the OFDM signal burst used in this modem consists of one detection preamble, one synchronization preamble, and several ZP-OFDM data transmission blocks. The purpose for the detection preamble is to trigger the receiver for data processing. The functions of the synchronization preamble in the form of one cyclic-prefixed (CP) OFDM block include synchronization and conveying a limited amount of control information, which includes the transmission mode and the length of guard zeros for the ZP-OFDM data blocks to be followed. This facilitates the AMC operation as the
receiver uses the correct mode parameters to decode the ZP-OFDM data blocks automatically when the transmitter switches to a different transmission mode.

3.2.2.3 AMC Procedure

Fig. 16 shows the AMC procedure to be tested in this thesis. First, a transmitter (TX) node will initiate an request-to-send (RTS) message, and the receiver (RX) node will reply clear-to-send (CTS) after receiving RTS. The CTS will report the channel conditions, upon which the TX node will choose a suitable transmission mode and transmit a data burst to the RX node.
3.2.3 Performance Metrics

One critical component of an AMC system is to find an appropriate performance metric, based on which the transmitter can switch to a suitable transmission mode.

3.2.3.1 Existing Performance Metrics

Input SNR (ISNR) is calculated in the time domain by comparing the received signal strength to the noise strength. Based on the time domain input-output Equation (12), after synchronization, the signal and noise portions can be distinguished. Denote $y_\nu[n]$ as the received sample at the signal portion and $w_\nu[n]$ as the received sample without signal transmission at the $\nu$-th hydrophone. The input SNR for the $\nu$-th hydrophone is calculated as:

$$\text{ISNR}_\nu = \frac{E\{ |y_\nu[n]|^2 \} - E\{ |w_\nu[n]|^2 \}}{E\{ |w_\nu[n]|^2 \}}.$$ (35)

Input SNR is proportional to the received signal power. However, input SNR may not reflect the performance of OFDM in underwater channels well since it does not capture the possible inter-carrier interference (ICI) due to Doppler effects.

Pilot SNR (PSNR) defined in the frequency domain has been used as a measure for an underwater OFDM system in [111]. Based on the frequency domain input-output Equation (13), the energy at the pilot subcarriers can serve as the indicator of useful signal strength, while the energy at null subcarriers will represent the noise strength, where the intercarrier interference is included as part
of noise. The pilot SNR at the \( \nu \)-th hydrophone is

\[
\text{PSNR}_\nu = \frac{E_{k \in S_\nu} \{ |z_\nu[k]|^2 \} - E_{k \in S_N} \{ |z_\nu[k]|^2 \}}{E_{k \in S_N} \{ |z_\nu[k]|^2 \}}.
\] (36)

Pilot SNR captures the ICI effect, however, it is computed before the channel estimation module. Given the same amount of pilot subcarriers, channels with a small number of paths could be estimated more accurately than those with a large number of paths. Hence, PSNR might not be a consistent performance metric as it ignores the impact of channel estimation to the system performance.

### 3.2.3.2 Proposed Performance Metric

In this work, we propose to use an effective SNR (ESNR) as the performance metric. The ESNR is computed after the receiver has successfully decoded a message, which is used to probe the channel condition and initiate the AMC operation. Based on the equivalent channel input-output relationship after MRC (17), the ESNR is calculated as:

\[
\text{ESNR} = \frac{E_{k \in S_D} \left[ |\hat{H}_\text{mrc}[k]s[k]|^2 \right]}{E_{k \in S_D} \left[ |z_{\text{mrc}}[k] - \hat{H}_\text{mrc}[k]s[k]|^2 \right]}.
\] (37)

Notice that the average is taken over the set of data subcarriers when calculating the ESNR, and \( s[k] \) is known since it is done after successful decoding of the probe message.
3.2.3.3 Simulation Results

We first simulate the system performance as a function of ISNR, PSNR and ESNR, respectively. Channel models are multi-path channel shown in (9) and (10), corresponding to cases of all paths have the same or different Doppler scale factors. The inter-arrival time of paths follows an exponential distribution with a mean value of 1 ms; this means that for two consecutive paths \( p \) and \( p + 1 \), the difference between the delays \( \tau_p \) and \( \tau_{p+1} \) follows an exponential distribution with mean value \( E[\tau_{p+1} - \tau_p] = 1 \) ms.

Three channel settings are used:

- **Setting 1**: The channel has \( N_{pa} = 10 \) paths, and the common Doppler scale \( a \) is zero;

- **Setting 2**: The channel has \( N_{pa} = 20 \) paths, and the common Doppler scale \( a \) is zero;

- **Setting 3**: The channel has \( N_{pa} = 20 \) paths, and the Doppler scale for each path is drawn randomly from a uniform distribution \( \alpha_p \sim U(-10^{-4}, 10^{-4}) \).

The time-varying channel introduces ICI.

All the decoding performances are based on \( N_r = 4 \) hydrophones. Input SNR and PSNR are calculated based on one hydrophone, while ESNR is calculated after combining signals from 4 hydrophones. Fig. 17 shows the block error rate as a function of different SNRs. The performance curves are inconsistent among three settings when input and pilot SNRs are used. On the other hand, the
performance curves are close to each other in different channel conditions when ESNR is used. This suggests that ESNR is a more robust performance metric for AMC mode switching.

![BLER graphs for different modes and settings](image)

**Figure 17**: BLER for five transmission modes.

### 3.2.3.4 Performance Results with Recorded Data

In addition to simulation results, performance results based on experimental data sets are presented here to further verify that ESNR is a consistent performance metric in different channel conditions. Three experimental settings are as follows.
• **Swimming pool:** These data sets are collected at a swimming pool at the University of Connecticut. The pool has a standard size of $25 \times 50$ square meters. Fig. 18(a) presents one estimated channel scattering function, which shows two-dimensional view of the channel (time and frequency). From Fig. 18(a) we can observe that channel length is about 20 ms. There are dense multipath arrivals due to the bounces from different sides of the pool.

• **Sea test (3 km and 5 km distances):** At Dec. 15, 2012, an experiment was carried out in the Long Island Sound, United States. Two sets of data corresponding to 3 km distance and 5 km distance were collected. Fig. 18(b) shows one estimated channel scattering function corresponding to the 3 km distance, while Fig. 18(c) depicts one estimated channel scattering function corresponding to the 5 km distance. In both cases, the channels contain only one cluster.

Gaussian noise has been artificially added to the recorded data sets in order to generate BLER points corresponding to different SNRs, as shown in Fig. 19. Note that in the 3 km sea test, the mode 5 transmission was not carried out, while in the 5 km sea test, the mode 1 transmission was not carried out. Fig. 19(c) shows that the BLER-vs-ESNR performance curves obtained from different channel environments are close to each other (less than 1 dB difference) when the same
transmission mode is used. This is not the case in Fig. 19(a) for the BLER-vs-ISNR curves. The BLER-vs-PSNR curves in Fig. 19(b) are also consistent in these experiments, probably because the channel conditions are relatively stable with little Doppler effect as shown in Fig. 18. However, inconsistency has been shown in Fig. 17(b) in simulation results.

We underscore that the objective in this section is not to explore how the BLER curves would shift under different SNR definitions, but rather to figure out which SNR can predict the BLER performance consistently in different channel conditions. Due to the consistency of the BLER-vs-ESNR curves as shown here,
we adopt the ESNR as the performance indicator for the AMC-OFDM field test in Section 3.2.4.

3.2.4 Real Time AMC Tests

A sea experiment was carried out in the South China Sea, near Kaohsiung City, Taiwan, from May 22 to May 28, 2013. Three nodes were involved in the AMC tests in May 22, 2013, as shown in Fig. 20. The distance between node 5 and node 7 was about 1038 m, and the distance between node 5 and node 9 was about 2188 m. The water depth was about 26 meters, and the modems
were about six meters below the surface. Fig. 21 shows the estimated channel scattering functions, where we can observe that the channel lengths were around 10 ms, with several multipath arrivals. In the channel from node 7 to node 5, some late arrivals have non-negligible Doppler shifts.

![Figure 20: Locations of the nodes used in the AMC tests.](image)

![Figure 21: Estimated channel scattering functions in the AMC tests.](image)

The test procedure follows that plotted in Fig. 16. In the experiment, each data burst consisted of 5 OFDM data blocks. The power levels of 50%, 25%, 12.5%, 6.2%, 3.1% and 1.5% were cycled through, so the power interval between two consecutive power levels was around 3 dB. We set the ESNR thresholds for
the 5 transmission modes to be: 3.8, 5.0, 7.4, 9.2 and 12.2 dB. They are set according to the BLER-vs-ESNR curves in Fig. 17(c), with the target BLER to be $10^{-2}$, plus a 2 dB protection margin.

During the experiment, there were unexpected impulsive noises, and almost all the data sets were affected with different degrees of severity. Fig. 22 shows one typical received waveform that was affected by impulsive noise. This is the major reason that leads to limited values in ESNR and no transmission of modes 4 and 5 even with increased transmission power.

![Waveform affected by impulsive noise.](image)

Figure 22: Waveform affected by impulsive noise.

Table 4: AMC test result (between nodes 9 and 5)

<table>
<thead>
<tr>
<th></th>
<th>50%</th>
<th>25%</th>
<th>12.5%</th>
<th>6.2%</th>
<th>3.1%</th>
<th>1.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1 - success</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>10</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>Mode 1 - failure</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mode 2 - success</td>
<td>42</td>
<td>62</td>
<td>51</td>
<td>4</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>Mode 2 - failure</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>Mode 3 - success</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Mode 3 - failure</td>
<td>9</td>
<td>4</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$R_e$ (kb/s)</td>
<td>1.48</td>
<td>1.64</td>
<td>1.85</td>
<td>1.18</td>
<td>1.38</td>
<td>1.30</td>
</tr>
<tr>
<td>Total number of blocks</td>
<td>65</td>
<td>80</td>
<td>55</td>
<td>20</td>
<td>35</td>
<td>15</td>
</tr>
</tbody>
</table>
3.2.4.1 Test 1: AMC Between Nodes 9 and 5

In this test, node 5 worked as the RX node, while node 9 worked as the TX node. Table 4 summarizes the test performance corresponding to different transmission power levels. In Table 4, $R_e$ is the equivalent data rate, calculated as:

$$R_e = \frac{\sum_{i=1}^{5} N_{bl,i}N_{bi,i}}{N_{tr}T_{bl}}$$  \hspace{1cm} (38)$$

where $N_{bl,i}$ is the successfully decoded data blocks in mode $i$; $N_{bi,i}$ is the bits per block in mode $i$, calculated in Table 3; $N_{tr}$ is the total number of transmitted
data blocks; $T_{bl}$ is one OFDM block duration (including guard zero length). Here in this test, guard zero length is set to be 150 ms, so $T_{bl} \simeq 320$ ms.

Fig. 23 shows the SNR performance according to different transmission power levels. Note that for Fig. 23 (a) and (b), each circle point corresponds to one calculated SNR (input SNR or PSNR) value in one OFDM block. For Fig. 23(c), each star point corresponds to the ESNR value for one RTS, and each circle point corresponds to the ESNR value for the first OFDM block in the data burst. Because there are 5 data blocks in one data burst, there are much fewer circle or star points in Fig. 23 (c) than in (a) and (b). Also in Fig. 23(c), ESNR thresholds used to distinguish different working modes have been plotted, which helps to show the distribution of AMC working modes.

### 3.2.4.2 Test 2: AMC Between Nodes 7 and 5

Table 5: AMC test result (between nodes 7 and 5)

<table>
<thead>
<tr>
<th>Power level</th>
<th>50%</th>
<th>25%</th>
<th>12.5%</th>
<th>6.2%</th>
<th>3.1%</th>
<th>1.5%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>29</td>
<td>74</td>
<td>49</td>
<td>41</td>
<td>28</td>
<td>32</td>
</tr>
<tr>
<td>Failure</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Mode 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Success</td>
<td>36</td>
<td>37</td>
<td>27</td>
<td>10</td>
<td>19</td>
<td>9</td>
</tr>
<tr>
<td>Failure</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$R_c$ (kb/s)</td>
<td>1.42</td>
<td>1.20</td>
<td>1.25</td>
<td>0.98</td>
<td>1.20</td>
<td>1.07</td>
</tr>
<tr>
<td>Total number of blocks</td>
<td>70</td>
<td>120</td>
<td>80</td>
<td>60</td>
<td>50</td>
<td>45</td>
</tr>
</tbody>
</table>

This test used node 5 as the RX node, and node 7 as the TX node. Table 5 presents the test results corresponding to different transmit power levels. Fig. 24 shows the SNR performance in this test. The input SNRs are generally higher in test 2 than those in test 1 due to the fact that the distance between RX and TX is shorter in test 2 than in test 1. The PSNR and ESNR are relatively stable and
remain at a low level in test 2, which explains the relative lower equivalent data rate in test 2 as compared to test 1.

Figure 24: SNR performance for AMC test 2.

From the performance results of tests 1 and 2, we have the following observations. As the transmission power increases, input SNR and pilot SNR increase. However, the ESNR does not increase much as the transmission power increases, due to the existence of impulsive noise, and the limitation of non-perfect channel estimation. This is why transmission modes in both tests kept almost stable no matter how transmit power varied. Although this scenario with impulsive noises is unexpected prior to the experiment, the AMC procedure has been successfully
carried out, which also shows that ESNR is an effective performance indicator for AMC mode switching.

### 3.2.5 Energy Consumption Optimized AMC

Section 3.2.4 focuses on an AMC system with the objective to maximize the data rate. In this section, still with the five modulation and coding pairs in Table 2, we focus on designing an AMC system aiming to minimize the energy consumption used for transmitting a data packet with a finite number of bytes. The so-called “green communications” concept has been popular with radio communication [27, 21, 11, 38], and here we explore it in underwater acoustic communications. This is important as most of the underwater acoustic communication modems use batteries as the power source, maintaining communication while using minimum energy consumption will prolong the operation time.

![Figure 25](image)

(a) Power level at 3%

(b) Power level at 25%

Figure 25: The measured instantaneous power consumption at different transmit power levels.
3.2.5.1 Problem Formulation

Here we use the power measurements from AquaSeNT’s OFDM modems [3] as the study example. As described in [26] and [103], AquaSeNT’s OFDM modems have been adopted to set up the community Ocean Testbeds for Underwater Networks Experiments (Ocean-TUNE). Our study here, in particular the power measurement results, would be useful for the users of the OCEAN-TUNE testbeds.

The goal is to minimize AquaSeNT’s OFDM modems’ energy consumption when transmitting a finite amount of information bytes; note that the numerical values might change as newer versions of modems become available. For AquaSeNT’s OFDM modems, the transmit power level can be adjusted by the user, as a percentage of the maximum allowed power. Fig. 25 shows the power consumption distribution for different transmit stages at two different power levels. From Fig. 25, we can categorize the power consumption during transmission to be the following five stages:

1. Overhead stage, in which the power consumption $P_{\text{overhead}}$ corresponds to the process of turning on the power amplifier.

2. Stage of transmitting guard zeros, in which the power consumption $P_{gz}$ corresponds to transmitting guard zeros.

3. Stage of transmitting detection preamble, in which $P_{\text{det}}$ is used to transmit detection preamble.
4. Stage of transmitting synchronization preamble, in which $P_{sync}$ is consumed to transmit synchronization preamble.

5. Stage of transmitting data blocks, in which $P_{sym}$ is consumed to transmit OFDM data symbols.

Note that the signal structure has been shown in Fig. 15.

Denote $L_i$ ($i = 1, 2, 3, 4, 5$) to be the number of bytes that can be transmitted per data block for each modem, which can be referenced in Table 3. For a total of $N_B$ information bytes to be transmitted, a total of

$$N_i = \left\lceil \frac{N_B}{L_i} \right\rceil$$  \hspace{1cm} (39)

data blocks are needed. If mode $i$ is chosen and the transmission power level is set at $x$ percent, then the total energy consumption is

$$E_{total}(i, x) = E_{overhead} + P_{gz}(T_{ini} + N_i T_{bl}) + P_{det}(x)T_{det} + P_{sync}(x)T_{sync} + P_{sym}(x)N_i T_{sym}$$  \hspace{1cm} (40)

where $P_{det}(x)$, $P_{sync}(x)$, and $P_{sym}(x)$ are the transmission power corresponding to the detection preamble, the synchronization preamble, and the OFDM data blocks at $x$ percent transmission power, respectively; $P_{gz}$ corresponds to the power consumption during the transmission of guard zeros; $T_{ini}$ corresponds to the initial guard zeros transmission time before data block transmission and $E_{overhead}$ corresponds to the overhead to start the power amplifier.
If a relationship between $P_{\text{sym}}(x)$ and the average block error rate performance can be found, then the optimization problem can be formulated:

$$
i_{\text{opt}} = \arg \min_{i} E_{\text{Total}}(i, x) \quad \text{subject to } \overline{\text{BLER}} \leq \text{BLER}_0 \quad (41)
$$

$$i \in \{1, 2, 3, 4, 5\}$$

Subject to certain block error rate requirement, the mode which minimizes total energy consumption will be selected.

The challenge of energy optimized AMC lies in how to determine the relationship between the transmission power and the receiver performance. Note that only the input SNR is related to the transmission power directly. However, the system performance does not depend on input SNR alone, as environment conditions affect the system performance considerably. Hence, for a particular system setup, prior knowledge needs to be obtained to establish the relationship between the BLER and the input SNR. This is in contrary to the effective SNR which provides a consistent performance metric for rate optimization.

Next we construct illustrative numerical examples assuming certain ISNR thresholds for mode switching.

### 3.2.5.2 Numerical Examples

Table 6 lists the related parameter settings for this example, concrete values for related stages shown in Fig. 25 are given out. The target BLER in this numerical example is set to be $10^{-2}$. 
Table 6: Energy consumption related parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{overhead}}$</td>
<td>0.986 J</td>
</tr>
<tr>
<td>$T_{\text{ini}}$</td>
<td>0.57 s</td>
</tr>
<tr>
<td>$T_{\text{sym}}$</td>
<td>170 ms</td>
</tr>
<tr>
<td>$T_{\text{bl}}$</td>
<td>220 ms</td>
</tr>
<tr>
<td>$P_{\text{gz}}$</td>
<td>5.812 W</td>
</tr>
<tr>
<td>$\text{BLER}_0$</td>
<td>$10^{-2}$</td>
</tr>
</tbody>
</table>

According to Fig. 25, we assume the ratio between energy consumption for transmitting detection preamble and synchronization preamble and energy consumption for transmitting one block OFDM data symbol to be a constant value. Based on measurement, $(P_{\text{det}}(x)T_{\text{det}} + P_{\text{sync}}(x)T_{\text{sync}})/(P_{\text{sym}}(x)T_{\text{sym}})$ is set to be 2.5.

In order to find out the relationship between receiver BLER and transmit power consumption, we first build up the relationship between BLER and receiver input SNR. As described above, this relationship depends on many other factors related to environmental conditions. Here we combine the information from the simulation results shown in Fig. 17(a), where channel setting 3 is chosen, and the sea test results shown in Fig. 19(a), where the four curves corresponding to the 3 km distance and the curve for mode 5 corresponding to the 5 km distance are chosen, to determine the approximate relationship between the BLER and ISNR. With the target $\text{BLER}_0$ set at $10^{-2}$ and adding a 2 dB protection margin, we choose the ISNR thresholds at the receiver side to be 3.3, 4.8, 9, 11.6, 18.4 dB for the five modes, respectively. Note that this is an idealistic assumption to construct the numerical examples. For a practical setup, the thresholds need to be determined by prior training.
Next, we find out the relationship between transmit power consumption and transmit power percentage, based on AquaSeNT’s OFDM modems. Fig. 26 shows the power consumption versus the transmit power level for AquaSeNT’s OFDM modems. Note that the measured power consumption is the summation of aforementioned guard zero transmission power \( P_{gz} \) and data block transmission power \( P_{sym}(x) \), which is shown as the curve with star marker. Removing the contribution of \( P_{gz} = 5.812W \), we get the curve that shows the relationship between the data block transmission power \( P_{sym}(x) \) and transmit power percentage \( x \), which corresponds to the curve with circle marked. We can decide the transmit power \( P_{sym}(x) \) at certain percentage \( x \) by looking up this curve, with piece-wise linear approximation between two consecutive measurements.

![Figure 26: Measured data block transmit power.](image)

At last, we use the input SNR plots shown in Fig. 23(a) and Fig. 24(a) for the Taiwan tests to build up the relationship between the transmit power percentage and the receiver input SNR. With this information, we have all the knowledge to infer required power consumption for certain receiver BLER. Corresponding
to the two tests with different distances, the following two energy optimization examples are constructed.

**Example 1.** For the test between nodes 9 and 5, line fitting for the ISNR points corresponding to the transmit power levels 3.1%, 6.2%, 12.5%, 25% and 50% in Fig. 23(a) leads to:

\[
\text{Input SNR} \approx 7.7845 \times \log_{10}(100x) + 6.1415 \text{ dB} \tag{42}
\]

Using (42), the transmit power levels corresponding to the required ISNRs of the five modes 3.3, 4.8, 9, 11.6, and 18.4 dB are 0.43%, 0.67%, 2.33%, 5.03% and 37.56%, correspondingly. With these parameters, Fig. 27(a) shows the energy consumption as a function of the number of information bytes for all the 5 modes. The selected optimal modes are shown in Fig. 27(b).

**Example 2.** With line fitting for the ISNR points corresponding to transmit power levels 1.5%, 3.1%, 6.2%, 12.5% and 25% in Fig. 24(a), we get the receiver ISNR as a function of the transmission power percentage, shown as the dashed
The energy consumed for all the five modes and the optimal modes are shown in Fig. 28(a) and (b), respectively.

These two numerical examples show that if the distance between the transmitter and receiver is small, the required transmission power corresponding to different modes does not vary much. The fixed energy overhead consumed by the power amplifier dominates. In such a case, the system can transmit with higher data rate modes in order to save the total number of blocks, and hence save the transmit energy. When the transmitter and receiver are far from each other, it requires much more power in order to increase the receiver SNR. In this case, transmitting in low data rate modes will be preferred.
3.2.6 Summary

This part of thesis presented an OFDM based AMC system for underwater acoustic communications based on a finite number of transmission modes. The effective SNR was proposed as a consistent performance metric for mode switching in different channel conditions. Simulation and experimental results validated the advantage of ESNR over other SNR definitions, and performance results of real time AMC operations in a sea environment were presented. At last, this part of study explored the AMC scheme based on energy consumption minimization for a finite-length data packet.
Chapter 4

DSP Implementation for Underwater Acoustic

OFDM

4.1 SISO and MIMO OFDM DSP Optimization for Underwater Acoustic Modems

4.1.1 Introduction to SISO and MIMO OFDM DSP Implementation for Underwater Acoustic Modems

As introduced in Section 1.4, there has been a growing interest in building distributed and scalable underwater wireless sensor networks (UWSN). For UWSNs, high performance and reliable physical layer underwater acoustic communications are essential. Recently, OFDM technique has received a great deal of attention and has been proven to be a viable option for underwater acoustic communications. The combination of OFDM and multi-input and multi-output (MIMO)
techniques has also been made to drastically increase the data rate through spatial modulation [55, 14, 30].

Compared with extensive theory development on underwater acoustic communications, there have been limited amounts of work on underwater acoustic communication modem development, especially for OFDM modems. As far as we know, there is no related work on underwater acoustic MIMO OFDM modem development.

In this part of thesis, we investigate the implementation of OFDM modems for underwater acoustic communications. First, we implement both single-input single-output (SISO) and multi-input multi-output (MIMO) OFDM modems on a floating-point TMS320C6713 DSP development board. Through the work load analysis, we optimize the implementation to accelerate the decoding speed. Real time operation has been achieved and the processing time for each OFDM block is much less than the block duration. We also examine the fixed-point implementation on a TMS320C6416 DSP development board for both SISO and MIMO systems. Running at a higher clock frequency (1 GHz) than TMS320C6713, the fixed-point implementation reduces the processing time by two thirds, with negligible performance degradation. Finally, we include some performance results with single transmitter and one or two receivers using data collected from the ACOMM10 experiment, conducted by Naval Research Laboratory (NRL) near New Jersey, July 2010.
4.1.2 OFDM Transceiver Design

The DSP board supports stereo audio input and output, and hence up to two channels are available for transmission and reception. Based on this hardware configuration, we consider three settings: (1) a SISO system, (2) a single-input multi-output (SIMO) system with one transmitter and two receivers, and (3) a MIMO system with two transmitters and two receivers, as shown in Fig. 29. For MIMO OFDM, two separate data streams are transmitted in parallel [55], while only one data stream is transmitted for SISO and SIMO OFDM [56]. The data format is illustrated in Fig. 1. The SISO transmission is obtained by turning the second transmitter off.

![Figure 29: The SISO, SIMO (1x2), and MIMO (2x2) system setups.](image)

4.1.2.1 Transmitter Design

The basic parameters for the zero-padded (ZP) OFDM are listed in Table 7. The sampling rate is \( f_s = 48 \text{ kHz} \) and the signal bandwidth is \( B = 6 \text{ kHz} \). The carrier frequency is \( f_c = 12 \text{ kHz} \), but can be changed easily to fit the transducer characteristics.
Table 7: ZP-OFDM Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling rate $f_s$</td>
<td>48 kHz</td>
</tr>
<tr>
<td>Center frequency $f_c$</td>
<td>12 kHz</td>
</tr>
<tr>
<td>Signal bandwidth $B$</td>
<td>6 kHz</td>
</tr>
<tr>
<td>OFDM block duration $T$</td>
<td>170.7 ms</td>
</tr>
<tr>
<td>Guard interval $T_g$</td>
<td>40 ms</td>
</tr>
<tr>
<td>Total block duration $T_{bl}$</td>
<td>210.7 ms</td>
</tr>
<tr>
<td>Subcarrier spacing $\Delta f$</td>
<td>5.86 Hz</td>
</tr>
<tr>
<td>Number of subcarriers $K$</td>
<td>1024</td>
</tr>
<tr>
<td>Number of data carriers $</td>
<td>S_D</td>
</tr>
<tr>
<td>Number of pilot carriers $</td>
<td>S_P</td>
</tr>
<tr>
<td>Number of null subcarriers $</td>
<td>S_N</td>
</tr>
</tbody>
</table>

As shown in Fig. 1, each data burst consists of two parts: the preamble and the data payload. The preamble is used for packet detection and synchronization. The data payload contains one or more ZP-OFDM blocks.

The preamble consists of two identical OFDM symbols with one CP as depicted in Fig. 1. Each OFDM symbol has $K_0 = 512$ subcarriers and the CP length is 60% of the OFDM symbol duration. Hence, the preamble has a duration of $(2 + 0.6)K_0/B = 221.9$ ms.

For the payload, each OFDM symbol has $K = 1024$ subcarriers, and hence the duration of one OFDM block is $T = K/B = 170.7$ ms. The ZP guard interval is of $T_g = 40$ ms, which can be changed depending on the channel delay spread. The total $K = 1024$ subcarriers are divided into $|S_N| = 96$ null subcarriers, $|S_P| = 256$ pilot subcarriers, and $|S_D| = 672$ data subcarriers. For MIMO-OFDM, the pilot subcarriers are split into two non-overlapping sets, one for each data stream [55].

QPSK constellation is used. We consider two channel codes:

- 64-state rate-1/2 convolution code, with the code generator as (133,171).
• rate-1/2 nonbinary LDPC code over GF(4) [44]. Each symbol over GF(4) is directly mapped to one QPSK constellation point.

Accounting for all the overheads of channel coding, null and pilot subcarriers, and guard intervals, the overall data rate is

\[
R = \begin{cases} 
\frac{1}{2} \cdot \frac{1.2|S_D|}{T + T_g} = 3.2 \text{ kb/s}, & \text{SISO-OFDM} \\
\frac{1}{2} \cdot \frac{2.2|S_D|}{T + T_g} = 6.4 \text{ kb/s}, & \text{MIMO-OFDM}
\end{cases}
\]  

(44)

4.1.2.2 Receiver Overview

Fig. 30 shows the procedure of the OFDM receiver.

Fig. 30 shows the procedure of receiver processing. When started, the device is in searching state. The codec keeps sampling the audio signal and searches for the high energy section which may indicate the beginning of a transmission. When the energy level of the signal samples reaches a threshold, the device enters the recording state, in which the modem incrementally saves samples for the duration of a transmission. The sampled data are first processed by the synchronization
module. If a valid preamble is detected, the receiver will begin data decoding for each OFDM block sequentially until the end of the data burst.

Next we will provide more details on the receiver synchronization and the payload block-by-block processing.

4.1.2.3 Synchronization

Synchronization can be done using either passband or baseband samples. Here, we follow the description in [66] and present the synchronization algorithm using baseband samples. Denote $y[n]$ as the baseband samples, a timing metric $M(d)$ is computed through a sliding window correlator [66, 113]:

$$M(d) = \frac{\sum_{m=0}^{K_0-1} [y^*(d + m) y(d + m + K_0)]}{\sqrt{\sum_{m=0}^{K_0-1} |y(d+m)|^2 \cdot \sum_{m=0}^{K_0-1} |y(d+m+K_0)|^2}}$$

(45)

where $d$ is the time index of the sliding window.

Each item of the metric $M(d)$ is calculated by multiplying two samples from two adjacent windows and at the same position, and then adding all the products up. Both windows then slide to the right by one sample to calculate the next item of $M(d)$ until it reaches the end of the signal.

In the absence of noise, $|M(d)|$ reaches a maximum magnitude within a plateau due to the preamble structure. The synchronization uses the middle of the plateau as the timing estimate to determine the starting position of the data signal. The middle of the plateau is calculated by finding the shoulders of this plateau (e.g. 90% of the maximum magnitude)[73]. Note that this procedure does not require any channel knowledge.
4.1.2.4 Payload Block-by-block Processing

After synchronization, the receiver accumulates the received samples. When one ZP-OFDM block is ready, the following tasks are carried out, as depicted in Fig. 30.

1. Downshift and low pass filtering (LPF). After downshifting to the baseband and low pass filtering, the received sequence is downsampled to the baseband rate.

2. Doppler shift compensation. The Doppler shift due to the channel variation is modeled as if it were due to carrier frequency offset (CFO) between the transmitter and the receiver. We compensate the CFO on the OFDM block for each tentative CFO $\epsilon$ and evaluate the FFT output on the null subcarriers. The energy of the null subcarriers is used as the cost function $J(\epsilon)$. An estimate of $\epsilon$ can be found through

$$\hat{\epsilon} = \arg \min_{\epsilon} J(\epsilon).$$ (46)

3. Channel estimation. For single-transmitter OFDM, all pilot subcarriers are used for one data stream. For MIMO OFDM, the pilot subcarriers are divided into two non-overlapping sets, with each data stream assigned one set of uniformly distributed pilot subcarriers [55]. The least-squares (LS) channel estimator is used for each transmitter and receiver pair, which amounts to two FFT operations [56].
4. Demodulation. For SISO OFDM, no demodulation step is needed. For SIMO OFDM, maximum ratio combining (MRC) is applied on each data subcarrier [56]. For MIMO OFDM, we use the zero-forcing (ZF) receiver in [55] to separate the two data streams, which amounts to inversion of a $2 \times 2$ matrix on each data subcarrier.

5. Channel decoding. Viterbi algorithm is applied for the decoding of convolutional code. The Min-Sum decoding is used for decoding of the nonbinary LDPC code [105].

To achieve real time operation in this setting, the receiver needs to decode a received OFDM block within a period of $T_{bl} = 210.7$ ms so the current block is processed before the next one arrives.

4.1.3 Floating-Point Implementation

We now describe the implementation results on a development board for the TMS320C6713 floating-point DSP, which runs at 225 MHz.

4.1.3.1 Hardware Platform

TMS320C6713 belongs to the TMS320C6000 DSP family, which uses VLIW, a very-long-instruction-word (VLIW) architecture [89]. It is among the most powerful floating-point DSPs from the TMS320C6000 family. The DSP core has eight independent functional units, including two fixed-point ALUs, four floating-/fixed-point ALUs, and two multipliers. It can execute up to eight instructions
simultaneously in one cycle. It has two general-purpose register files, which have 32 32-bit registers in total [88]. It provides 256 Kbytes on-chip L2 memory which can be configured as unified cache or SRAM.

The board has a plenty of peripherals that facilitate the development. Each DSP board is equipped with one AIC23 codec, one enhanced direct memory access (EDMA) controller, two multichannel buffered serial ports (McBSPs). In our implementation, one serial port McBSP0 is configured as a unidirectional port to control and configure the AIC23 codec. The codec receives serial commands through McBSP0, which set configuration parameters such as volume, sampling rate and data format. Audio data are transferred back from the codec through the other serial port McBSP1, which is configured as a bidirectional serial port. The enhanced DMA controller (EDMA) is configured to take every 16-bit signed audio sample arriving on McBSP1 and store it in a buffer in memory. Once the buffer is full, the EDMA controller generates an interrupt to inform the DSP core that the audio data are ready for processing. The board also has 4 Mbytes external dynamic RAM (DRAM), which is sufficient for our implementation.

4.1.3.2 Implementation

The majority of the implementation efforts is on optimization, which can be classified into two categories: algorithm optimization and programming optimization. For the *algorithm optimization*, we adopt proper algorithms for those
computationally intensive operations, such as convolution and CFO estimation. For example, the following steps have been taken.

- Use the overlap-add and FFT-based algorithm to perform linear convolution. This well-known method can speed up significantly linear filtering operations.

- Use adaptive searching algorithm for CFO estimation. Instead of using a simple linear search, we use adaptive search to reduce the total number of evaluations of the cost function. The basic idea of the adaptive search is as follows. First, a coarse searching (with large interval) is performed to construct a subset of the candidate CFOs in which the target CFO will reside with high probability. Then a bi-sectional search is applied for quick convergence.

- Recursively compute the auto-correlation. Instead of involving all samples inside the sliding window as in (45), the recursive method calculates all correlation items except for the first one using only one sample at the beginning and one entry at the end, as described in [66].

For the *programming optimization*, we apply a few coding strategies, such as loop unrolling and compiler directives, to speed up the processing. Also, we efficiently use the hardware resources on the DSP chip such as VLIW architecture and on-chip memory. For example, we use single-precision (32-bit) floating-point, instead of double precision (64-bit), to represent data. This change leads to
shorter execution time because (1) single-precision floating-point operations are faster than double-precision operations, and (2) with a smaller data size, more data entries can be kept in the on-chip L2 memory, hence reducing the external memory access overhead. We also carefully manage the on-chip L2 memory, using part of it as L2 cache and part of it for scratchpad memory managed by application code.

We profile the implementation and identify time consuming steps. Whenever possible, we utilize the DSP library from TI to reduce the development time and provide better portability. We also hand-wrote some routines, e.g., scaling of complex vectors, in assembly language, which are not available in TI DSP library.

4.1.3.3 Processing Time

Table 8: Processing time of one SISO OFDM block of 210 ms

<table>
<thead>
<tr>
<th></th>
<th>Conv. code, 64-state [ms]</th>
<th>LDPC, GF(4) [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downshifting, LPF</td>
<td>12.61</td>
<td>12.61</td>
</tr>
<tr>
<td>CFO estimation</td>
<td>12.20</td>
<td>12.20</td>
</tr>
<tr>
<td>Channel estimation</td>
<td>0.84</td>
<td>0.84</td>
</tr>
<tr>
<td>Decoding</td>
<td>9.80</td>
<td>21.55</td>
</tr>
<tr>
<td>Other</td>
<td>2.52</td>
<td>2.52</td>
</tr>
<tr>
<td>Total per-block time</td>
<td>37.97</td>
<td>49.72</td>
</tr>
</tbody>
</table>

We use the 32-bit timer on the DSP board to measure the execution time. The per-block processing time of one OFDM block of duration $T_{bl} = 210$ ms is shown in Table 8 for SISO, and in Table 9 for MIMO settings. It takes about twice the time to decode in the MIMO case as that in the SISO case as two channels are processed to recover two data streams simultaneously.
Table 9: Processing time of one MIMO OFDM block of 210 ms

<table>
<thead>
<tr>
<th></th>
<th>Conv. code, 64-state [ms]</th>
<th>LDPC, GF(4) [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Downshifting, LPF</td>
<td>25.14</td>
<td>25.14</td>
</tr>
<tr>
<td>CFO estimation</td>
<td>24.34</td>
<td>24.34</td>
</tr>
<tr>
<td>Channel estimation</td>
<td>1.42</td>
<td>1.42</td>
</tr>
<tr>
<td>ZF equalization</td>
<td>2.4</td>
<td>2.4</td>
</tr>
<tr>
<td>Decoding</td>
<td>19.90</td>
<td>43.56</td>
</tr>
<tr>
<td>Other</td>
<td>3.79</td>
<td>3.79</td>
</tr>
<tr>
<td>Total per-block time</td>
<td>76.99</td>
<td>100.65</td>
</tr>
</tbody>
</table>

With either convolutional or nonbinary LDPC codes, real-time decoding is achieved. The per-block-processing time is much less than the block duration, which allows more functionalities to be implemented in the future. For channel decoding alone, it takes more than twice the time for the nonbinary LDPC code than the convolutional code. Using LDPC instead of convolutional coding, the overall processing time per OFDM block increases by 30%, however, the performance is considerably enhanced as shown next.

Figure 31: The experimental setup in a water tank.
4.1.3.4 BLER Performance

To compare the performance of convolutional and LDPC codes, we recorded 10 data bursts from the experiment setup in a water tank as depicted in Fig. 31, where each data burst has 10 ZP-OFDM blocks as shown in Fig. 32. In the MIMO setup with two transmitters and two receivers, there exist four channels. The channel impulse responses measured on one receiver are plotted in Fig. 33, corresponding to two transmitters. The channels have delay spread about 10 ms, meaning that the last path is longer than the first path by about 10 ms × 1500 m/s = 15 m. Hence, there are significant reflections going on within the water body of about 2 × 0.5 × 0.5 m³ in the water tank.

Figs. 34 and 35 show the BLER performance of SISO and MIMO settings. The definition of PSNR is given out in (36). For each point plotted, at least a total of 1000 blocks are decoded, with at least 10 different noise realizations added.
to those 100 recorded blocks. The two data streams in Fig. 35 have different performance due to different channel strengths as shown in Fig. 33. At BLER of $10^{-2}$, nonbinary LDPC codes outperform convolutional codes by about 2 dB, which agrees with the simulation results in [44]. Such a performance improvement justifies the use of nonbinary LDPC codes despite the increase of the decoding time.
4.1.4 Fixed-Point Implementation

The DSP for fixed-point implementation is TMS320C6416. Like TMS320C6713, it belongs to the TMS320C6000 family and has a VLIW architecture [89]. TMS320C6414 can run at much higher clock rates than floating-point DSPs so it can reduce the execution time of many applications. In addition, the fixed-point implementation can facilitate future FPGA or ASIC implementations where fixed-point operations are adopted for smaller area and lower energy consumption. On the other hand, compared to the floating-point implementation, the fixed-point implementation may suffer from some loss on the error-rate performance, which needs to be investigated.

Most features on TMS320C6416 are similar to those on TMS320C6713. The improvements include higher clock rate (1 GHz), larger on-chip L2 memory (1 Mbytes), and more registers (64 32-bit registers in total).

4.1.4.1 Implementation

The major difference between the fixed- and floating-point implementations is the representation of values. Most optimizations, especially those at the algorithm level, are the same in both implementations.

Since the size of registers on TMS320C6416 is 32 bits, we use 32-bit fixed-point numbers. To make the data compatible with the C64x DSP library, we adopt the Q.31 format, where the most significant bit is the sign bit, and the remaining 31 bits represent the fraction part of a number. The Q.31 format can only represent
numbers between $[-1, 1)$. Consequently, proper scaling is required during the processing. For example, the sampled input signals are naturally normalized to $(-1, 1)$.

The added complexity of fixed-point implementation is due to data scaling operations for all the processing modules to avoid data overflow, e.g., FFTs and inverse fast Fourier transforms (IFFTs). Scaling factors are carefully chosen to prevent significant precision loss. The most important one is data-dependent dynamical scaling for each OFDM block before Viterbi decoding, as the log-likelihood ratio computation involves one multiplication per data subcarrier for the SISO case and several multiplications per data subcarrier in the MIMO case [55].

4.1.4.2 Processing Time

The processing time is recorded in Table 10. We can see that the fixed point implementation is about three times faster than floating point implementation. This reduction is significant. Note that the scaling operations add some extra computation time, which are not significant. On the other hand, the fixed-point DSP is running at 1 GHz, about 4.5 times faster compared to the floating point DSP at 225 MHz. Note that the execution time cannot be simply scaled with the clock rate. These two DSPs are different in many aspects. The operation type and the latency are different. For example, the fixed-point DSP does not support the multiplication of Q.31 format natively, which has to be done through a sequence of instructions. On the other hand, the floating-point DSP uses one
instruction to perform the multiplication of two single precision floating-point numbers. In addition, the size of the register files and the internal memory are different.

Table 10: Processing Time of Floating- and Fixed-point implementation

<table>
<thead>
<tr>
<th></th>
<th>SISO OFDM</th>
<th>MIMO OFDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating-point</td>
<td>38.06 ms</td>
<td>76.65 ms</td>
</tr>
<tr>
<td>Fixed-point</td>
<td>12.69 ms</td>
<td>25.14 ms</td>
</tr>
</tbody>
</table>

4.1.4.3 BLER Performance

To evaluate the performance loss due to fixed-point implementation, we add noise directly to the recorded passband data. All modules in the block-by-block processing in Fig. 30 are involved in the fixed-point implementation. Here we use the convolutional code for illustration.

![Figure 36: The comparison of fixed- and floating-point implementation of SISO-OFDM.](image)

(a) Pilot SNR vs input SNR

(b) BLER vs input SNR

Fig. 36 shows the pilot SNR and block error rate (BLER) performance of SISO OFDM at different input SNR levels, and Fig. 37 shows the counterparts
for MIMO-OFDM. The differences between fixed- and floating-point implementations are negligible. Since the input data are sampled with 16 bits, the 32-bit fixed-point processing has enough room to avoid overflow without dropping meaningful bits. Certainly, programming effort is much increased for fixed-point implementation as compared to the floating-point alternative.

![Graph](image1.png)  
(a) Pilot SNR vs input SNR

![Graph](image2.png)  
(b) BLER vs input SNR

Figure 37: The comparison of fixed- and floating-point implementation of MIMO-OFDM, 2 transmitters and 2 receivers.

One could pursue fixed-point implementation and see how the receiver performance changes as the number of bits decreases, e.g., from 32 bits to 24 or 16 bits. While this is interesting and important for future low power FPGA implementation, this is not within the scope of this thesis. Since the decoding time is much less than the block duration, our next immediate goal is to incorporate more advanced processing algorithms, such as sparse channel estimation via compressive sensing techniques [7, 46] or iterative processing for progressive intercarrier interference mitigation [45].
4.1.5 Performance Results of Replaying Data from a Sea Test

In this section, we test the modem performance based on data collected from the ACOMM10 experiment conducted by the Naval Research Laboratory, July 2010. This experiment was done in an open sea, about 100 miles from the sea shore, near New Jersey. The water depth was about 70 m. The transmitter and receiver were placed about 50 m below the surface, and the distance between them was about 3 km. One transmitter was used, and there were eight phones on a receiver array to receive the signal. The same signal set as described in Section 4.1.2 was used, except that the center frequency is changed to $f_c = 18$ kHz.

Here, we use the data from two phones (index 3 and 4 on the array) to illustrate the SISO and SIMO performance. The recorded data sets are replayed through a desktop to the DSP modem, directly connecting the audio output from the laptop to the audio input of the DSP modem. We here report the float-point decoding performance to illustrate several effects with real data. Obvious, the per block processing time remains the same as that reported in Section 4.1.3. Note that we expect negligible performance loss if 32-it fixed-point implementation is used as the recorded data is of 16-bit, the same situation as the tests based on the data from the water tank.
4.1.5.1 Received SNR Fitting

In this data set, there are a total of 22 transmissions. Every two consecutive transmissions use the same transmit power, therefore they can be regarded as one group. For each group, a 3 dB transmit power reduction was applied from the previous group. Specifically, let $P_{\text{init}}$ denote the transmit power in dB for the first group. Then, the transmit power of the $i$th group is $P_{\text{init}} - 3(i - 1)$, with the last (11th) group having power $P_{\text{init}} - 30$ dB.

Fig. 38 plots the estimated received input SNRs (calculated as (35)) corresponding to the 22 transmissions. Knowing that the transmit power decreases by 3dB for consecutive groups, we fit a line with a slope of -3dB to calibrate the SNR computation, based on the least squares (LS) formulation. After calibration, the received SNRs corresponding to the 11 groups of transmissions are: 28.3, 25.3, ...., -1.7 dB.
4.1.5.2 Channel Impulse Response and Synchronization

Fig. 39 plots sample channel impulse responses of this environment. The delay spread is about 40 ms, with many strong paths. Sample synchronization plateaus are shown in Fig. 40, from files with different received SNRs.

Figure 39: Estimated channel impulse responses on two phones.

Figure 40: The correlation plateau for synchronization.
4.1.5.3 Decoding Results

Fig. 41 shows the floating point DSP decoding results of these 11 groups of transmissions. As mentioned before, in each group, there are two transmissions. In each transmission, there are 60 blocks. So for each group, there are in total 120 blocks. Fig. 41 presents the percentage of successfully decoded blocks, using one phone alone and using two phones with maximum ratio combining (MRC).

![Graph showing decoding performance](image)

Figure 41: Decoding performance of phone 3 and 4.

The BLER performance of combined decoding has significant improvement over single phone decoding, and the performance of LDPC coding outperforms that of convolutional coding significantly in both single-phone and two-phone cases. This observation is consistent with the results from Section 4.1.3.

Also for a system with strong channel codes, there is a “water-fall” region on the performance. When SNR is above a certain threshold, all the blocks can be decoded, while the BLER performance deteriorates quickly when the SNR is below a certain threshold. The transition from near error-free to undecodable
performance happens quickly, e.g., within 2 dBs as shown in Fig. 34; Further simulation results on LDPC codes can be found in [44].

4.1.5.4 Discussions

In this test, the sampled data from real data set are directly fed into the DSP modem to evaluate the core receiver processing performance. This is a significant step toward in-situ testing of a practical modem. To come up with a practical modem ready to be deployed, the following components need to be worked on: automatic gain control (AGC) circuits at the receiver, power amplifier and matching circuits for long-distance transmission, water-proof packaging of the DSP modem as well as good connections with transducer and hydrophones. Those issues are under current consideration.

One issue identified by replaying the ACOMM10 data sets is that the energy-based detection might fail for signals with low received SNRs, although the data are still decodable after combining signals from multiple phones. Hence, a better detection mechanism is needed to prevent detection failure before decoding failure. In our future work, we plan to add a frequency-sweep preamble based on a hyperbolic or linear frequency modulated (HFM/LFM) waveform. Matched filtering at the receiver increases the effective signal energy at the detection point, and hence enables robust detection.
4.1.6 Summary

In this section of thesis, we reported implementation results of SISO and MIMO OFDM acoustic modems using both floating- and fixed-point DSP platforms. Real-time decoding was achieved in all the considered cases. Nonbinary LDPC coding improves the performance relative to convolutional coding, at the cost of increased decoding time. Fixed-point implementation drastically reduces the processing time compared with the floating-point counterpart, running at a higher clock frequency. The floating point decoding performance for SISO and SIMO OFDM was also tested on a data set collected in open sea.

The implementation effort in this thesis provides useful insights toward realizing a practical OFDM modem for high-data-rate and reliable underwater acoustic communications, which in turn will facilitate research and development of underwater networked systems.

4.2 System Implementation of OFDM-DCC in Underwater Acoustic Channels

4.2.1 Introduction to Dynamic Coded Cooperation in Underwater Acoustic Communications

Relay-based strategies provide a new dimension to the design space of wireless networks in which the coverage and throughput may be significantly enhanced [106, 2, 57]. In a network consisting of three nodes denoted as
source (S), relay (R), and destination (D), there are various relay strategies: amplify-and-forward (AF), decode-and-forward (DF) and compression-and-forward (CF) [2, 57]. Out of all the choices, the dynamic coded cooperation (DCC) scheme is particularly interesting [16, 53, 47]. In DCC scheme, the half-duplex relay listens until it can decode the message correctly and then switches to the transmission mode. When transmitting, the relay superimposes its transmission on the ongoing transmission from the source to the destination. There is no extra transmission time scheduled for the relay, making it bandwidth efficient. The source can be even unaware of the existence of the relay. In underwater acoustic communications, a surface buoy can serve as a relay to assist the communication between any pair of underwater nodes, as shown in Fig. 42. Because the surface buoy can be solar-powered, having unlimited power supply, and can have large processing capability with a large receiver array, it can improve the communications among resource-limited underwater nodes. Currently, there are a few relay strategies targeting for underwater acoustic cooperative communications, e.g., distributed space time block coding in [94], and amplify forward and decode forward in [39, 12, 101, 13, 22, 77]. Unlike DCC, these cooperative schemes require extra transmission time scheduled for the relay after the source transmission.

Existing works on the DCC scheme [16, 53, 47] assume frequency-flat channels, and assume symbol level (or sample level) synchronization of the transmissions from the source and the relay. Underwater acoustic channels, however, have very
large delay spreads. Also, it is hard to synchronize the transmission on the sample level from distributed nodes. This part of the thesis will describe the design and implementation of a practical OFDM modulated dynamic coded cooperation scheme for underwater acoustic communications. Repetition Redundancy (RR) strategy and layered inter-block erasure-correction coding and intra-block error-correction coding are adopted. The most appealing feature of this RR cooperation scheme is that both source and destination nodes can be unaware of the existence of a relay. As long as the signal from source and relay nodes achieve block level synchronization, receiver node will treat these two signals as one signal experiencing a composite multipath channel.

![Surface relay system](image)

**Figure 42:** Surface relay system.

### 4.2.2 OFDM Modulated Dynamic Coded Cooperation

We consider a three-node network consisting of a source, a destination and a relay which helps the transmission from the source to the destination. All
the transceivers work in a half-duplex fashion, which is the case in underwater acoustic networks. The channels among the source, the relay, and the destination are multipath fading channels with large delay spread. For this reason, OFDM modulation is adopted. We assume that the guard interval between consecutive OFDM symbols, either cyclic prefix (CP) or zero-padding (ZP), is larger than the maximum delay spread plus the offset between the signals from the source and the relay to reach the destination. As such, there is no inter-block interference (IBI) between consecutive OFDM blocks at the relay and at the destination.

The source divides a packet into multiple blocks, say $N_{bl}$ blocks, with each block modulated on one OFDM symbol. The source transmits $N_{bl}$ OFDM blocks, which will reach both the relay and the destination. Note that these blocks are inter-block coded through erasure-correction coding, while each of them is intra-block coded through Nonbinary LDPC code. The details about this layered coding scheme will be introduced next.

4.2.2.1 Layered Erasure- and Error-Correction Coding

In many existing designs of underwater OFDM transmissions, channel coding has been applied on a block-by-block basis [56], which is also the case for the OFDM modem implemented in [111]. Without altering the channel coding performed within each OFDM block, a separate layer of erasure correction coding can be applied across OFDM blocks [9]. Rateless coding has been suggested in [16, 53], to enable the DCC operation. However, rateless coding requires a large
number of blocks, which might not be suitable for underwater acoustic communications. Hence, we suggest the following approach to use erasure-correction coding over a finite number of blocks.

We use nonbinary linear coding to perform the inter-block erasure-correction coding. Operating over GF($2^8$), every eight bits are group into one byte before encoding. First, divide one packet into $I_{bl}$ information blocks, where each block contains $P$ symbols in GF($2^8$). Denote the $p$-th symbol of the $i$-th block as $b[i; p]$. An encoder which generates $N_{bl}$ coded symbols from $I_{bl}$ information symbols is applied as

\[
\begin{pmatrix}
  c[1; p] \\
  \vdots \\
  c[N_{bl}; p]
\end{pmatrix}
= \begin{pmatrix}
  1 & 1 & \cdots & 1 \\
  1 & \alpha & \cdots & \alpha^{I_{bl}-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & \alpha^{N_{bl}-1} & \cdots & \alpha^{(N_{bl}-1)(I_{bl}-1)}
\end{pmatrix}
\begin{pmatrix}
  b[1; p] \\
  \vdots \\
  b[I_{bl}; p]
\end{pmatrix},
\]

where $\alpha$ is a primitive element in GF($2^8$) [58]. After the erasure-correction coding, each set of $P$ symbols \{c[l; 1], c[l; P]\} will be forwarded to the error-correction channel encoder to generate the coded symbols to be modulated in the $l$-th OFDM block.

The OFDM blocks which fail in channel decoding will be discarded; (each block has its own CRC flags, as done in e.g., [111]). Thanks to the Vandermonde structure of the code generation matrix in (47), any square submatrix drawn from it is guaranteed to be nonsingular and thus invertible in the finite field. Hence, as long as the relay collects $I_{bl}$ correctly decoded blocks, all the information symbols
can be recovered, and the whole packet can be regenerated. Note however, that this layered decoding approach is expected to be suboptimal relative to a joint decoding approach where all accumulated blocks are decoded jointly.

4.2.2.2 Relay Operations

The relay has two operational phases: the listening phase and the cooperative transmission phase. First, the relay is in the listening phase. For every new OFDM block that it collects, the relay tries to decode the whole packet using the accumulated OFDM blocks. After successfully decoding the transmitted packet before the end of the transmission from the source, the relay switches to the cooperation phase.

Denote $N_{li}$ as the number of OFDM blocks that the relay has used for successful decoding. The relay starts to superimpose its transmission to the ongoing transmission from the source, from the $(N_{li} + \Delta + 1)$-th block to the $N_{bl}$-th block, where $\Delta$ is an integer to be determined. In this thesis, with repetition redundancy (RR) cooperation scheme, the relay regenerates and transmits identical OFDM blocks as the source, from the block index $(N_{li} + \Delta + 1)$ to $N_{bl}$.

During the cooperative transmission phase, the OFDM blocks from the source and the relay need to be aligned at the block level at the receiver side. This is achieved through a delay control mechanism at the relay. Define $T_{sr}$, $T_{rd}$ and $T_{sd}$ as the transmission delays between the source and the relay, the relay and the destination, and the source and the destination, respectively. Denote the start
time of the $N_{li}$-th block at the source as $t_0$, the relay processing time as $T_{\text{proc}}$, and the relay waiting time as $T_{\text{wait}}$. To synchronize the reception of the $(N_{li}+\Delta+1)$-th OFDM block at the destination, the following relationship should be satisfied as illustrated in Fig. 43:

$$t_0 + T_{\text{sr}} + T_{\text{bl}} + T_{\text{proc}} + T_{\text{wait}} + T_{\text{rd}} \approx t_0 + (\Delta + 1)T_{\text{bl}} + T_{\text{sd}}.$$  \hfill (48)

Hence, the extra waiting time prior to the cooperative transmission at the relay is:

$$T_{\text{wait}} \approx \Delta T_{\text{bl}} - T_{\text{proc}} - (T_{\text{sr}} + T_{\text{rd}} - T_{\text{sd}}).$$  \hfill (49)

The parameter $\Delta$ should be taken as a small integer that leads to a nonnegative waiting time $T_{\text{wait}}$. The processing time $T_{\text{proc}}$ is known to the relay. The difference $(T_{\text{sr}} + T_{\text{rd}} - T_{\text{sd}})$ depends on the source-relay-destination geometry. In a favorable geometry where $(T_{\text{sr}} + T_{\text{rd}} - T_{\text{sd}})$ is small and with a relay having $T_{\text{proc}} < T_{\text{bl}}$, the value of $\Delta$ could be as small as one.
The relay needs to have the knowledge of the source-relay distance \( d_{sr} \), the relay-destination distance \( d_{rd} \), and the source-destination distance \( d_{sd} \) to determine the waiting time from (49). Since the acoustic modems are often equipped with the ranging functionality, the relay needs to probe the source and the destination to obtain \( d_{sr} \) and \( d_{rd} \). The source needs to probe the destination to obtain \( d_{sd} \), and conveys it to the relay. For example, the source-destination distance could be put into the packet header, along with the source ID and destination ID.

The cooperation phase lasts for \( N_{bl} - N_{li} - \Delta \) OFDM blocks, hence the duration is dynamic depending on the channel quality from the source to the relay.

4.2.2.3 Receiver Processing at the Destination

Let \( N_r \) denote the number of receiving elements at the destination. We consider an OFDM modulation with \( K \) subcarriers. For the first \( N_{li} + \Delta \) blocks, the destination only receives the transmission from the source. After the necessary pre-processing steps which involves Doppler compensation and FFT operation [56], the input-output relationship of the \( l \)-th received OFDM block at the \( \nu \)-th receiving element is:

\[
z_{\nu}[l] = H_{\nu,sd}[l]s[l] + \xi_{\nu}[l], \quad \nu = 1, \ldots, N_r, \quad l = 1, \ldots, N_{li} + \Delta,
\]

where \( z_{\nu}[l] \) is the vector containing the frequency measurements across \( K \) subcarriers, \( s[l] \) is the vector of transmitted symbols on \( K \) subcarriers, \( H_{\nu,sd}[l] \) denotes
the channel mixing matrix for the channel between the source and the destination, and $\xi_{\nu}[l]$ is the ambient noise.

For the last $N_{bl} - N_{li} - \Delta$ blocks, the destination receives the superposition of the signals from the source and the relay.

$$
z_{\nu}[l] = (H_{\nu,sd}[l] + H_{\nu,rd}[l]) s[l] + \xi_{\nu}[l],
= H_{\nu,\text{equ}}[l] s[l] + \xi_{\nu}[l], \quad \nu = 1, \ldots, N_r, \quad l = N_{li} + \Delta + 1, \ldots, N_{bl}
$$

where $H_{\nu,rd}[l]$ denotes the channel mixing matrix for the channel between the relay and the destination. Clearly, an equivalent channel, which consists of multipath arrivals from both the source and the relay, is formed.

The destination can adopt a receiver as in [56] that ignores the residual intercarrier interference (ICI) after Doppler compensation, with the assumption that the channel mixing matrices $H_{\nu,sd}[l]$ and $H_{\nu,rd}[l]$ become diagonal. Or the destination can adopt a receiver as in [45] that deals with ICI explicitly imposing a banded structure on the channel mixing matrices.

### 4.2.2.4 Discussions

In this RR cooperation scheme, the relay transmission increases the received power and provides multipath diversity benefits to the last $N_{bl} - N_{li} - \Delta$ OFDM blocks. No change is needed at the destination. Note that the OFDM modem in [111] performs channel estimation on a block-by-block basis, in order to deal with fast channel variations in underwater environments. Hence, the receiver does
not need to be aware of the existence of a relay. Further, instead of one relay, multiple relays can be easily added into the OFDM-DCC scheme if using the RR cooperation.

4.2.3 Implementation of One OFDM-DCC System

We have implemented the OFDM-DCC scheme with layered coding and RR cooperation into the modem prototype [111], which adopts ICI-ignorant receiver with least-square based channel estimator. More implementation details can be found in Section 4.1. For this specific system, the modem parameters are set as shown in Table 11.

<table>
<thead>
<tr>
<th>Table 11: Experiment related parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center frequency $f_c$:</td>
</tr>
<tr>
<td>Sampling frequency $f_s$:</td>
</tr>
<tr>
<td>Bandwidth $B$:</td>
</tr>
<tr>
<td>FFT size:</td>
</tr>
<tr>
<td># of data subcarriers $</td>
</tr>
<tr>
<td>Time duration:</td>
</tr>
<tr>
<td>Guard interval in ZP-OFDM:</td>
</tr>
<tr>
<td>Modulation:</td>
</tr>
<tr>
<td>Coding:</td>
</tr>
</tbody>
</table>

Each OFDM block carries 80 bytes of payload data. Here, we set $I_{bl} = 8$ and $N_{bl} = 18$ for the erasure-correction coding, and hence each packet has 640 bytes of information data.

Following are the two major tasks that have been done in implementation:

- Erasure-correction decoding. Gaussian elimination is used for matrix inversion over the finite field for erasure-correction decoding. Thanks to the
small code length, a very small computational overhead is added to the
modem processing. Specifically, it only takes 0.111 ms to decode one code-
word with \( I_{bl} = 8 \) symbols in \( GF(2^8) \), using the TI DSP chip TMS320C6747
[111]. To decode the packet of 640 bytes, the total time is only 8.88 ms,
much smaller than the OFDM block duration.

- **Synchronization.** To achieve the block-level synchronization as described in
  Section 4.2.2.2, two changes have been made to the modem: 1) the relay
  performs a fine synchronization step to locate the starting time of each
  OFDM block that it has received [8]; and 2) After correctly decoding the
  packet, the relay needs to hold on its transmission for \( T_{\text{wait}} \) seconds. A timer
  is issued, and when it expires, a hardware interrupt is triggered that will get
  the transmission of the OFDM blocks actually started. This way, the relay
  can align its transmission to achieve the block-level (quasi-) synchronization
  for the OFDM blocks received at the destination from both the source and
  the relay.

### 4.2.4 Experiment Results of the Implemented System

In this section, experiment results of the implemented OFDM-DCC system
in both swimming pool and sea tests are presented.
4.2.4.1 Swimming Pool Test

The experiment was carried out in Aug. 9, 2012, in the Brundage Pool at the University of Connecticut, with the setup shown in Fig. 44. With the source node and destination node set in two sides of the pool, the relay node was put in three different locations in the middle, as shown in Fig. 45. The distance from the source node S to the destination node D is about \( d = 50 \text{ feet} \). Relay node was placed between the source node and the destination node. According to its distance to the source node S, three possible relay locations were included, \( d/4 \), \( d/2 \) and \( 3d/4 \) away from the source node, respectively. In all the three settings, the relay node has the same transmit power as the source node. Since the source, relay, and destination are on a line, \( T_{sr} + T_{rd} - T_{sd} = 0 \). The value of \( \Delta \) is set to be one in this experiment as \( T_{proc} < T_{bl} \).

Figure 44: Experimental setup in the swimming pool.
A total of 4 scenarios were tested: no relay, and one relay at three different locations. In each scenario, 40 packets of data were recorded at the destination node D, where each packet has $N_{bl} = 18$ OFDM blocks with inter-block erasure-correction coding as specified in Section 4.2.2.1. The input SNRs as measured at the received blocks without relay are high, e.g., about 20 dB. Fig. 46 shows the channel statistics of one packet from the scenario of relay at $d/4$ away from the source node. Clearly, the RR cooperation leads an equivalent multipath channel that is stronger than the original multipath channel through signal superposition.
We now add white Gaussian noise of different levels to the 40 recorded packets in each test scenario. Fig. 47 plots the packet error rate (PER) performance in different test scenarios, as a function of the variance of the added noise which is normalized by the variance of the recorded ambient noise in the signal band. Note that during this experiment, the source node has enough transmission power so that the decoding performance at the relay is similar in all locations, with $N_{li} = 8$. Since the noise is only added at the recorded data set at the destination locally, the closer the relay node to the destination, the better the PER performance becomes due to the higher SNRs for the OFDM blocks received at the cooperation phase. This trend is clearly observed in Fig. 47.

Figure 47: The packet error rate is obtained by adding noise to the recorded data at the destination. Note that the relay operation is done online in real time.
Figure 48: The locations of the source (node 4), the relay (node 5) and the destination (node 9).

4.2.4.2 Sea Test

The Underwater Sensor Network (UWSN) Lab at University of Connecticut participated a joint experiment led by the National Sun Yat-sen University, at the sea near the Kaohsiung City, Taiwan, May 22-28, 2013. The OFDM-DCC experiment was carried out on May 26, 2013, where the source, the relay, and the destination were deployed as shown in Fig. 48. Using the ranging function of the modems, the reported distances are: $d_{sr} = 1.63$ km, $d_{rd} = 2.39$ km, and $d_{sd} = 3.72$ km. The water depths at the source, relay, and destinations were about 27, 26, and 22 meters, respectively. The OFDM modems were attached to the surface buoys, at a water depth of 6 meters. One OFDM modem and one surface buoy during the deployment are shown in Fig. 49.

The OFDM-DCC firmware from the swimming pool test was loaded to the OFDM modems deployed in this experiment. A total of 189 transmissions were
transmitted, and each transmission contained 20 zero-padded OFDM blocks encoded using (49) with $I_{bl} = 8$ and $N_{bl} = 20$. The block delay was set as $\Delta = 4$ during the experiment.

The waveform of one data set recorded at the destination is shown in Fig. 50, where the received signals were much stronger during the relay cooperation phase. Note also that there existed impulsive noises, which would affect the communication performance for those affected blocks. Both the relay and the destination decoded the received blocks online. The performance results are as follows.

- Due to the short distance to the source, the relay decoded the data very well. In 188 transmissions, the relay was able to decode the whole packet with the first eight received blocks, and in one transmission, the relay used 9 blocks to decode the packet.
Figure 50: One received waveform after bandpass filtering; there are some impulsive noises.

- The destination kept decoding 20 OFDM blocks for each transmission. For each block index from 1 to 20, define the block error rate as the ratio of the number of erroneous blocks to the total number of transmissions. As shown in Fig. 51 (a), the BLER is around 0.06 before the relay cooperation, and it decreases to around 0.02 after the relay cooperation, when averaged over all 189 transmissions. The high BLER is likely due to the impulsive noise. After excluding 11 transmissions with a large number of block errors, the BLER averaged over the remaining 178 transmissions is around 0.025 before the relay cooperation and is around 0.01 after the relay cooperation. The pilot signal to noise ratio (PSNR), defined as the signal power at the pilot subcarriers to the power at the null subcarriers, is shown in Fig. 51
Figure 51: OFDM-DCC sea test results.

(b), averaged over 189 transmissions. A 2.5 dB increase is observed after the relay cooperation.

Fig. 52 shows the estimated channels before and after the cooperation. For the composite channel after relay cooperation, the first cluster corresponds to the channel from the source to the destination, and the second cluster corresponds to the channel from the relay to the destination. It can be seen that there is a 15-millisecond gap between the peaks of these two clusters, reflecting the synchronization offset.

In short, this is a successful demonstration of the OFDM-DCC operation in a sea environment. With the RR strategy, the relay improves the performance of the source to destination communication without introducing any changes to the transmission procedure between the source and the relay.
4.2.5 Summary

In this part of thesis, we introduced an OFDM modulated dynamic coded cooperation (DCC) scheme for underwater relay networks with long multipath channels, which is based on repetition redundancy (RR) strategy and layered erasure- and error-correction coding. The proposed OFDM-DCC system has been implemented on an underwater acoustic communication modem prototype, and experiments in a swimming pool and in a recent sea test were carried out to demonstrate the real-time operation in a three-node network.

The proposed OFDM-DCC scheme is especially appealing for underwater acoustic networks where some powerful relay nodes can be used to assist each communication between a source and a destination.
Chapter 5

Field Performance of Underwater Acoustic OFDM

5.1 Introduction to the OFDM Modem Deployment in Chesapeake Bay

There have been growing interests in building up underwater acoustic communication systems and networks [17, 80, 79, 40, 25, 23]. While some applications require short-time responses, many applications of underwater acoustic sensor networks require long term deployment and monitoring. The most recent and influential long term ocean monitoring project is the Ocean Observatories Initiative (OOI), which aims to build up networked infrastructure of sensor systems to measure the physical, chemical, geological and biological variables in the ocean and seafloor [68].
In addition to large scale open water observatories, there are a number of observing buoys in shallow water, near shore and in bays and rivers. One example is NOAA’s Chesapeake Bay Interpretive Buoy System (CBIBS), a network of 11 buoys at various positions in the bay, each of which gathers meteorological, oceanographic, and water quality data. The information from each buoy is wirelessly transmitted to shore, and is made available via NOAA’s web site, http://buoybay.noaa.gov.

The CBIBS “smart buoys” deliver real-time data on weather, water conditions, and water quality, as summarized in Table 12.

Table 12: Data provided by CBIBS

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Surface Water Temp [°C]</strong></td>
<td>Surface Sal</td>
<td>Surface Turb NTU</td>
<td>Surface DO Sat [%]</td>
</tr>
<tr>
<td><strong>Bottom Water Temp [°C]</strong></td>
<td><strong>Bottom Salinity</strong></td>
<td><strong>Bottom Turb NTU</strong></td>
<td><strong>Bottom DO Sat [%]</strong></td>
</tr>
<tr>
<td><strong>Bottom Water Dep [ft]</strong></td>
<td>Wave Max Ht [ft]</td>
<td>Wave Sig Ht [ft]</td>
<td>Wave Mean Ht [ft]</td>
</tr>
<tr>
<td>Wave Mean Per Sec</td>
<td>Wave Dir Deg</td>
<td>Current Vel [mm/s]</td>
<td>Current Dir Deg [°M]</td>
</tr>
</tbody>
</table>

In addition to surface measurements, water quality measurements at the bay floor provide key insights into the dynamic processes that drive the sustainability of Chesapeake bay. One of NOAA’s 11 CBIBS buoys has installed a bottom node, a YSI Sonde 6600 multivariable sensor, that hourly measures the following: chlorophyll, dissolved oxygen, oxygen saturation, acidity, salinity, turbidity, water depth (with effective tide measurement), conductivity, and temperature.

In many established semi-permanent observatories with bottom nodes, project planners have opted to use wireless transmission of data from the seafloor to the surface buoy. In the case of the CBIBS, like others, the data from the bay
floor is transmitted using an acoustic modem. This configuration affords several advantages, chiefly among them the elimination of failed cables due to harsh weather, and the avoidance of maintenance, installation, and decommissioning costs associated with hard-wired designs.

But on the other hand, although acoustic modems have been used for many years, there are some well-documented technical challenges regarding the reliability of acoustic modem signal reception. Specifically, the well-known multipath problem caused by acoustic echoes can greatly harm communication quality. Frustratingly, the dynamic conditions of the underwater environment can be unstable, resulting in perfectly good communication for a period of time, followed by a period of poor or completely failed communication. This makes troubleshooting difficult, and makes the choice of cabled vs wireless design less clear cut.

In March 2012, NOAA deployed an OFDM-based acoustic modem to transmit back the sensored data from the bottom node of the Gooses Reef site. The location of this buoy is shown in Figure 53, with the global positioning system (GPS) coordinates to be (38.5563 N, 76.4147W). Online data can be found at http://buoybay.noaa.gov/locations/goosesreef.

Each hour, the seafloor mounted YSI Sonde 6600 delivers a packet of encoded information regarding the nine key parameters, and the acoustic modem attempts to transfer this data to the topside modem, mounted in the buoy itself. Through an acknowledgement system, the two modems recognize a successful or failed
This deployment in CBIBS is the first practical application of the AquaSeNT OFDM modem [3]. Since March 2012, the modems have been running for about two years, sending sensor data on a hourly basis. The modems also record all the received raw waveforms into an internal storage space in a cyclic fashion, with the latest files replacing the oldest files. During one maintenance, the data sets over a two-month period, April 8, 2013 to June 5, 2013, were retrieved from the modem.

The rate of successful communication events has been quite good, over 80%, but this is not perfect. The following part of this thesis will describe the detailed analysis of the data, correlate failed communication with the exceedingly challenging multipath environment of the Gooses Reef Buoy, and propose several data
analysis and decoding steps that would improve performance to achieve success rates of about 97% at the expense of receiver complexity and processing delay.

5.2 Online Receiver Performance

5.2.1 Modem Signalling Format

Two OFDM modems were used in this deployment. One modem was deployed as bottom node, while the other modem was deployed in the surface, attached to a buoy. Slant range between modems varied from 10 to 50 meters as the buoy moved on its mooring chain. Due to the short distance, the power was set at 10% transmission power; the performance is not expected to be limited by the transmission power, but rather by the channel-induced self distortion.

During this deployment, the following simple activities were carried out by the two nodes in every hour:

1. The bottom node will transmit a data packet to the surface node.

2. Upon receiving the block, the surface node will reply with an acknowledge-ment (ACK) signal.

3. If the bottom node receives the ACK, all activities for this hour are over.

   If 20 seconds have passed and no ACK signal has been received in TX, the system will go to step 1 and repeat. If these 3 steps have been repeated for 3 times, all activities for this hour are over.

These processes are summarized in Fig. 54.
The signal structure is as shown in Fig. 15, with 3 ZP-OFDM data blocks following the preambles. The preamble of length about 0.5 second is primarily used for detection and synchronization, and carrying some control information, while the following 3 ZP-OFDM blocks carry data information. Each block has OFDM duration $T = 170.7$ ms and guard interval $T_g = 150$ ms, which was conservatively set to be much larger than the channel length. Mode 1 of BPSK constellation and rate 1/2 coding in Table 2 is used, which carries a payload of 38 bytes per block. Hence, the packet of three blocks carries 114 bytes of sensor data as payload. Since LDPC channel coding is applied to each ZP-OFDM data block, the receiver can decide whether decoding for each block succeeded or not through parity check at the decoder. If and only if all the 3 data blocks’ decoding succeeded, the whole data file is regarded as decoding succeeded. Otherwise, it’s regarded as packet error.
5.2.2 Online Receiver Performance

![Graphs showing SNR and PSNR distributions over time]

Figure 55: Online modem performance; plotted based on the log files

During the two month deployment, 1310 valid files were collected at the receiver side, which include 221 files that failed in decoding at the receiver modem. The modems collected the decoding outputs into a log file, which includes the time of reception, the estimated Doppler speed, the input signal to noise ratio (SNR), the pilot signal to noise ratio (PSNR), and others. The input SNR and PSNR are defined as shown in (36) and (35).

Based on the log files from the surface modem and the environmental parameters from the surface buoy, we plot the SNR distributions over the two-month period, where the packets in success and the packets in failure are marked differently. From Fig. 55, we have the following observations.

1. There are large temporally dynamics, where the system SNR values were constantly changing within a big range all the time. For example, the
dynamic range for ISNR were 5 to 40 dB, while PSNR were about 2 to 25 dB.

2. Almost all the decoding failed data sets have relative low PSNR values, as expected, where there are a few files failed with high ISNR values.

We now explore the question what environmental factors might be related with these dramatic SNR changes. Fig. 56 plots the relationship between the PSNR values, the wind speed and the maximum wave height from May 6 to May 16. The values of PSNR, wind speed and maximum wave height have been smoothed and normalized in order to better present their relationship. From Fig. 56, we can clearly observe a negative correlation between wind speed, maximum wave height and PSNR values: at those times when wind speed and maximum wave height are large, system PSNR values are small, and vice versa.

![Figure 56: Factors related with PSNR](image)

(a) PSNR vs. wind speed  
(b) PSNR vs. maximum wave height

Figure 56: Factors related with PSNR

With a close look at Fig. 56 (a), one can find that the changes of PSNR value lag behind the changes of wind speed a little bit. As shown in Fig. 56 (b),
the changes of PSNR values are more consistent with the change of wind speeds. This is further verified by Fig. 57, which shows that the change of maximum wave height is basically consistent with that of wind speed, except there is a lagging effect. Similar SNR vs. wind speed relationship and the lagging effect between wind speed and wave height have also been illustrated in [32].

![Figure 57: Relationship between wind speed and maximum wave height](image)

The fact that some blocks have high input SNRs but low PSNRs suggests that the intercarrier interference due to Doppler effect might be severe. So advanced algorithms with explicit ICI mitigation are expected to improve the receiver performance, which will be explored next.

### 5.3 Offline Data Processing

As stated in Section 5.2, there were 221 out of 1310 data files failed during the online operation in the two-month period. In this section, we adopt advanced receiver algorithms to process those 221 data files offline. The objective is to
see how much performance improvement can be achieved with advanced receiver algorithms, without consideration of complexity and real-time implementation issues.

5.3.1 ICI-Ignorant Iterative Receiver

After initial Doppler compensation and time-to-frequency conversion, the online modem operation assumes the following channel input-output relationship in the frequency domain at the $\nu$-th receive element as

$$z_{\nu}[m] \approx H_{\nu}[m]s[m] + \eta_{\nu}[m],$$

(52)

where $s[m]$ is the transmitted symbol on the $m$-th subcarrier, $z_{\nu}[m]$ is the frequency-domain observation on $m$-th subcarrier at the $\nu$-th element, $H_{\nu}[m]$ is the frequency channel response of the $m$-th subcarrier at the $\nu$-th element, and $\eta_{\nu}[m]$ is the frequency domain additive noise on the $m$-th subcarrier of the $\nu$-th element. Note that the equivalent noise $\eta_{\nu}[m]$ contains both the ambient noise and the residual intercarrier interference. With the model in (52), the ICI effect is ignored during the channel estimation, data detection, and channel decoding modules.

The online modem processing is non-iterative, where channel estimation, data detection, and channel decoding are carried out only once. To improve the performance, we first adopt an iterative receiver based on the system model in (52); such a receiver is termed as an ICI-ignorant iterative receiver [42], which iterates among the following steps.
1) Channel estimation. In the first round, channel estimation is carried out based on the pilot subcarriers on each receiver. In later rounds, the tentative decisions from the LDPC decoder are treated as additional pilots, and hence all the measurements on the pilot and data subcarriers are used for channel estimation.

2) Data detection. Maximum ratio combining is applied to combine the received signals from four hydrophones. Soft information in the form of LLR is then generated.

3) Channel decoding. Nonbinary LDPC decoding is used at the receiver, which declares a decoding success if all parity check equations are satisfied during the message passing iterations. With decoding failure, tentative decisions are made to be used for next round of channel estimation.

Fig. 58 shows that the typical channel delay spread for this deployment is around 20 ms. During the offline processing, we set the target channel length for channel estimation to be 20 ms. This side information will also help to improve the channel estimation accuracy.

We set the maximum number of iterations as 20. The iterative ICI-ignorant receiver is able to recover 66 failed data files successfully. Now there are 155 failed data files left.
5.3.2 ICI-Progressive Receiver

Fig. 59 shows a typical scattering function plot for one data file of this deployment. We clearly observe that different paths are associated with different Doppler speeds and the Doppler spread can be significant. A resampling operation can only remove the mean Doppler effect and the OFDM system will suffer from the inevitable ICI effect. Hence, it is worthwhile to look into underwater OFDM receivers that explicitly consider ICI.

Here we adopt the progressive ICI mitigation framework as proposed in [45]. At the frequency domain, the channel input-output relationship at the $\nu$-th receiver is

$$z_\nu[m] \approx \sum_{k=m-D}^{k=m+D} H_\nu[m, k]s[k] + \eta_\nu[m],$$

where $H[m, k]$ specifies how the symbol $s[k]$ on the $k$-th subcarrier contributes to the measurement on the $m$-th subcarrier and $D$ is termed as the ICI depth.
In this model, each symbol $s[k]$ generates interferences to $D$ neighbors to the left and $D$ neighbors to the right. Equation 53 includes Equation 52 as a special case when $D = 0$.

The ICI progressive receiver is essentially an iterative receiver. It starts with $D = 0$ and can iterate several times as in the ICI-ignorant iterative receiver. Then it can increase the ICI depth by one and continue the iterative processing. When $D > 0$, the channel estimation module search channel paths on a two-dimensional delay-Doppler plane, rather than on a one-dimensional delay line. We will use a linear minimum mean square error (MMSE) detection that combines the signals from four hydrophones to generate soft LLR information as needed by the channel decoder. Detailed descriptions of the progressive receiver framework can be found in [45].
With the use of the ICI progressive receiver where the maximum number of iterations are 20, 10, 5 and 2 for $D = 0, 1, 2, 3$, respectively, 16 more failed data files have been recovered. Now there are 139 remaining failed data files.

5.3.3 Data-driven Sparsity Factor Optimization and Effective Noise Based Multichannel Combining

The application of the ICI progressive receiver in Section 5.3.2 has followed the standard settings in [45]. We now look for performance improvement, and the two critical components we identified are the channel estimation module and multichannel combining module.

5.3.3.1 Data-driven Sparsity Factor Optimization

The sparse channel estimation at the $\nu$-th receiver is formulated as [7]

$$\hat{x}_\nu = \arg \min_{x_{\nu}} |A_{\nu} x_{\nu} - z_{\nu}|^2 + \zeta_{\nu} \|x_{\nu}\|_1$$

(54)

where $\| \cdot \|_1$ denotes the 1-norm, $x_{\nu}$ is the time domain channel response, $z_{\nu}$ is the frequency domain observations, $A_{\nu}$ is the dictionary matrix, $\zeta_{\nu}$ is the sparsity factor, which controls the sparsity of estimated channel. Define $\zeta_{\nu,\text{max}} = \|A_{\nu}^T z_{\nu}\|_\infty$, where $\| \cdot \|_\infty$ is the infinity norm. If $\zeta_{\nu} \geq \zeta_{\nu,\text{max}}$, the optimal solution to (54) is $\hat{x}_\nu = 0$. Hence, the sparsity factor $\zeta_{\nu}$ shall fall in the range of $[0, \zeta_{\nu,\text{max}}]$. Note that practical channels are not exactly sparse, but only approximately sparse. Also note that the sparsity of channel is not known prior to the deployment, and the channel changes over time. The processing in Section 5.3.2 has followed the
suggestion in [45] that
\[
\zeta_\nu = \frac{c_\nu}{\sqrt{\text{SNR}}} \zeta_{\nu,\text{max}}
\] (55)

where \(c_\nu\) was set to be 0.125, and SNR is the estimated effective SNR.

How to set the sparsity factor is a critical issue. Due to the time varying nature of channel, it will be better to tune the sparsity factor according to instant channel condition. In Section 3.2.3.2, the concept of effective SNR has been proposed as the performance indicator of an adaptive modulation and coding (AMC) system. The distinct feature of effective SNR is that it considers the impact of channel estimation quality on system decoding performance. Here, similarly, we propose to utilize the effective noise after channel estimation as the criterion for sparse factor tuning. Following the equivalent channel input-output relationship shown in (52) for the ICI-ignorant case, if we denote the estimated channel on the \(m\)-th subcarrier of the \(\nu\)th hydrophone to be \(\hat{H}_\nu[m]\), then the effective noise is defined as

\[
\chi_\nu[m] = z_\nu[m] - \hat{H}_\nu[m]s[m],
\] (56)

here, \(\chi_\nu[m]\) includes ambient noise, ICI, and channel estimation error. The variance of effective noise \(\chi_\nu[m]\) can be estimated on data subcarriers \(\mathcal{S}_D\), pilot subcarriers \(\mathcal{S}_P\), and null subcarriers \(\mathcal{S}_N\). According to the set of subcarriers on which the variance is estimated, we categorize the variances of effective noise into the following three sets:

- Estimated on data subcarriers \(\hat{\sigma}^{\nu,\mathcal{S}_D}_2 = E_{m\in\mathcal{S}_D} |\chi_\nu[m]|^2\).
• Estimated on pilot subcarriers $\hat{\sigma}_{\nu,P}^2 = E_{m \in S_p} |\chi_{\nu}[m]|^2$.

• Estimated on null subcarriers $\hat{\sigma}_{\nu,N}^2 = E_{m \in S_N} |\chi_{\nu}[m]|^2$.

The variance of effective noise estimated on all the subcarriers

$$\hat{\sigma}_{\nu,\text{all}}^2 = E_{m \in S_N \cup S_P \cup S_D} |\chi_{\nu}[m]|^2$$

is a weighted version of $\hat{\sigma}_{\nu,D}^2$, $\hat{\sigma}_{\nu,P}^2$, and $\hat{\sigma}_{\nu,N}^2$.

For the purpose of sparsity factor tuning, we aim to minimize the effective noise as seen on all the subcarriers:

$$\hat{c}_\nu = \arg \min_{c_\nu} \hat{\sigma}_{\nu,\text{all}}^2.$$ 

(58)

Note that the symbols on all the subcarriers are needed to evaluate $\hat{\sigma}_{\nu,\text{all}}^2$. When applying this technique to recover the failed data files, sparsity indexes will be first trained based on the recovered signal of one block decoded successfully in a data file. Then the optimized sparsity indexes will be applied to decode the failed data blocks in the same data file. Note that the tuning process is carried out independently on each phone.

Fig. 60 shows one training example of effective noise calculated on the data subcarriers $\hat{\sigma}_{\nu,D}^2$ vs. the sparsity index $c_\nu$ on all the 4 phones for data file F0017858. Here in this example, in order to better illustrate the idea, in the decoding process, the same sparsity index $c_\nu = c$ is chosen for all phones and the sparsity factor for each phone is calculated as (55). According to Fig. 60, the effective noise calculated on all subcarriers is a convex function of sparsity...
index. In this example of using one common sparsity index for decoding, the
decodable sparsity indexes for phone 2, 3 and 4 are those minimize the effective
noise. But for phone 1, the decodable region doesn’t lie in those sparsity indexes
that minimize the effective noise, as phone 1 has distinct pattern of the relation-
ship between sparsity index and effective noise. This example well illustrates the
necessity of training each phone independently, and applying different sparsity
indexes to different phones.

![Figure 60: The variance of the effective noise on data subcarriers as a function of the sparsity index $c_\nu$ for the data file F0017858](image)

**5.3.3.2 Effective Noise Based Weighting**

Fig. 60 shows that the variances of the effective noises measured on different
receiver hydrophones are different from each other, meaning different quality of
channel estimation on different hydrophones. In this work, we propose to utilize
the effective noise measured on pilot subcarriers $\hat{\sigma}_{\nu,P}^2$ as the initial weighting factor
when carrying out multichannel combining [56, 45]. In the iterative receivers,
after certain iterations, the estimated noise variance \( \hat{\sigma}_\nu^2 \) on the data subcarriers based on tentative symbol decisions will be utilized when calculating the weights for multichannel combining. This is in contrast to the modem online processing, where the noise variances \( \hat{\sigma}_{\nu,N}^2 \) at the null subcarriers were used to guide the multichannel combining due to the lack of knowledge of data symbols.

After applying the data-driven per-phone sparsity factor optimization and the effective noise based multichannel weighting, another 99 failed files have been recovered, which means there are only \( 139 - 99 = 40 \) failed data files left.

In summary, Fig. 61 shows the bar plot of success rate progress after each additional processing. The final success rate increases from 83.1% to 97.0%. Fig. 62 shows the final distribution of succeeded files, recovered files, and failed files.

![Figure 61: The bar plot on the packet success rates with different receiver processing](image)
Figure 62: The distribution of data files with different decoding results after offline processing.

5.4 Summary

AquaSeNT OFDM modems have been deployed in the CBIBS system in the Chesapeake Bay since March 2012. This section of thesis analyzes the performance based on the data recorded during a two-month period. Based on the log files, we analyzed the online decoding performance, and further correlated with environmental condition parameters (wind speed, wave height). It is shown that system SNR has strong positive correlation with decoding performance, while it has strong negative correlation with wind speed and wave height (wave height seems to be a lagging pattern of wind speed).

We then pursued the application of more advanced algorithms, including iterative ICI-ignorant receiver, ICI-progressive receiver with preset parameters, and with optimized parameters based on a new data-driven approach. A significant number of failed data files have been recovered, improving the packet success rate from 83.1% to 97.0%.
In short, the dynamic sea environment brings big challenges for underwater acoustic OFDM communication systems. Medium movement manifested as wave brings Doppler effect to underwater acoustic communications, which leads to ICI in OFDM system. If not appropriately dealt with, ICI will degrade OFDM system performance significantly. This thesis also proposed a data-driven approach to optimize the receiver parameters based on instant channel realizations.

Through this analysis, we have the following suggestions to improve the system performance.

- **Physical layer solution.** One can implement advanced receiver algorithms at the modem physical layer. Although real-time decoding might not be achieved, it is tolerable for certain types of applications, such as the CBIBS system at the Chesapeake Bay.

- **Link layer solution.** There are large temporal dynamics. It is not wise to pursue retransmissions right after a failed communication event. The retransmission shall happen in the next hour or next couple of hours. The idea is to wait the communication window after the ocean environment has significantly changed. A retransmission at a much delayed time is expected to substantially improve the link-layer performance, at the expense of data update time, which is tolerable in long-term monitoring applications.
Chapter 6

Conclusions

This dissertation focuses on algorithm design, DSP implementation and field performance for underwater acoustic OFDM. Our major contributions are as follows.

- We explored and compared different Doppler scale estimation methods in a specific underwater acoustic OFDM system.

- We designed and implemented a practical real time underwater acoustic system with adaptive modulation and coding.

- We implemented an OFDM modem prototype for underwater acoustic communication, based on TI DSP platforms.

- We implemented a three-node OFDM-DCC network based on a modem prototype, and carried out real time sea experiments.
• We analyzed the field performance of OFDM modem in a long term deployment, and correlated it with environmental factors. All the work in this dissertation are dedicated to bringing the underwater acoustic OFDM technology into practical systems.

• In aspect of algorithm design, the study on different Doppler scale estimation methods provides a good reference for practical underwater acoustic OFDM systems on the selection of a suitable estimation method. The work on AMC-OFDM successfully brings AMC technique into underwater acoustic OFDM systems. It builds the first real time underwater acoustic AMC-OFDM system.

• In DSP implementation, regarding to point to point communication, this dissertation implements real time SISO and MIMO OFDM underwater acoustic modem prototypes. Based on these prototypes, various underwater acoustic OFDM modems and networks are made available. Regarding to underwater acoustic communication networks, this thesis implements a three-node OFDM-DCC network. Real time sea test has proven the success of the implemented system.

• Through field performance analysis, this thesis evaluates the impact from environmental factors on underwater acoustic OFDM systems. The results from this part of thesis can serve as a guidance for future improvement of underwater acoustic OFDM systems.
Bibliography


[100] ——, “Adaptive modulation and coding for underwater acoustic OFDM,” 

mission in underwater acoustic networks,” in IEEE Symposium on Under-
water Technology, April 5-8 2011.

reception for OFDM in underwater acoustic communications,” IEEE Trans. 

“Long Island Sound testbed and experiments,” in Proc. of IEEE/MTS 

underwater sensor networks,” in Proc. of the ACM International Workshop 
on UnderWater Networks (WUWNet), Los Angeles, CA, Sep. 2006.

of LDPC codes over GF(q),” in Proc. of International Conference on Com-

ity based on code superposition,” in Proceedings of IEEE International 

amplitude and delay variations of underwater acoustic channels for block 
decoding of orthogonal frequency division multiplexing,” Journal of the 

underwater acoustic direct-sequence spread spectrum communications,” J. 

carrier underwater acoustic communication,” in 4th International Confer-
ence on Wireless Communications, Networking and Mobile Computing, 

power amplifiers for underwater acoustic OFDM transmissions,” in Proc. 

based receiver implementation for OFDM acoustic modems,” Elsevier Jour-


